4. Reset Control and Power Down in Arria II Devices

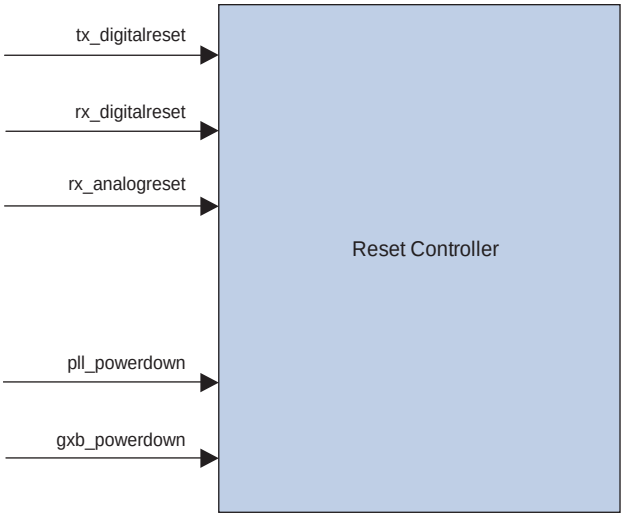
AIIGX52004-3.1

Arria® II GX and GZ devices offer multiple reset signals to control transceiver channels and clock multiplier unit (CMU) phase-locked loops (PLLs) independently. The ALTGX Transceiver MegaWizard™ Plug-In Manager provides individual reset signals for each channel instantiated in the design. It also provides one power-down signal for each transceiver block.

This chapter includes the following sections:

Figure 4–1 shows the reset control and power-down block for an Arria II GX or GZ device.

Figure 4–1. Reset Control and Power-Down Block



User Reset and Power-Down Signals

Each transceiver channel in the Arria II device family has individual reset signals to reset its physical coding sublayer (PCS) and physical medium attachment (PMA) blocks. Each CMU PLL in the transceiver block has a dedicated reset signal. The transceiver block also has a power-down signal that affects all the channels and CMU PLLs in the transceiver block.

All reset and power-down signals are asynchronous.

Table 4–1 lists the available reset, power-down, and status signals.

Table 4–1. Reset, Power-Down, and Status Signals for Arria II Devices (Part 1 of 2)

Signal	Description
Reset Signals For Each Transceiver Channel:	
tx_digitalreset *(1)*	Provides asynchronous reset to all digital logic in the transmitter PCS, including the XAUI transmit state machine, the built-in self test (BIST) pseudo-random binary sequence (PRBS) generator, and the BIST pattern generator. This signal is available in the ALTGX MegaWizard Plug-In Manager in **Transmitter Only** and **Receiver and Transmitter** configurations. The minimum pulse width for this signal is two parallel clock cycles.
rx_digitalreset *(1)*	Resets all digital logic in the receiver PCS, including the XAUI and GIGE receiver state machine, the XAUI channel alignment state machine, the BIST-PRBS verifier, and the BIST-incremental verifier. This signal is available in the ALTGX MegaWizard Plug-In Manager in **Receiver Only** and **Receiver and Transmitter** configurations. The minimum pulse width for this signal is two parallel clock cycles.
rx_analogreset	Resets the receiver CDR, receiver deserializer, and signal detect in the receiver buffer. This signal is available in the ALTGX MegaWizard Plug-In Manager in **Receiver Only** and **Receiver and Transmitter** configurations. The minimum pulse width is two parallel clock cycles. The busy signal has precedence over the rx_analogreset assertion. When the busy signal is high, rx_analogreset is ignored.
Power-Down Signal For Each CMU PLL in the Transceiver Block:	
pll_powerdown *(2)*	Each transceiver block has two CMU PLLs. Each CMU PLL has a dedicated power-down signal called pll_powerdown. The pll_powerdown signal powers down the CMU PLLs that provide high-speed serial and low-speed parallel clocks to the transceiver channels. Note: While each CMU PLL has its own pll_powerdown port, the ALTGX MegaWizard Plug-In Manager instantiation provides only one port per transceiver block. This port power downs one or both CMU PLLs (if used).
Power-Down Signal Common to the Transceiver Block:	
gxb_powerdown *(2)*	Powers down the entire transceiver block. When this signal is asserted, the PCS and PMA in all the transceiver channels and the CMU PLLs are powered down. This signal operates independently from the other reset signals
Status Signals:	
pll_locked	Indicates the status of the transmitter PLL. A high level on this signal indicates that the transmitter PLL is locked to the incoming reference clock frequency.
rx_pll_locked	A high level on this signal indicates that the receiver CDR is locked to the incoming reference clock frequency.
rx_freqlocked	Indicates the status of the receiver CDR lock mode. A high level indicates that the receiver is in lock-to-data mode. A low level indicates that the receiver CDR is in lock-to-reference mode.

Table 4–1. Reset, Power-Down, and Status Signals for Arria II Devices (Part 2 of 2)

Signal	Description
busy	When the busy signal is high during offset cancellation, the signal detect (in the receiver buffer), receiver CDR, and receiver deserializer are enabled regardless of the rx_analogreset signal state. However, the rx_pll_locked and rx_freqlocked signals always deasserted if rx_analogreset is asserted regardless of the busy signal.

Notes to Table 4–1:

(1) The tx_digitalreset and rx_digitalreset signals must be asserted until the clocks out of the transmitter PLL and receiver CDR are stabilized. Stable parallel clocks are essential for proper operation of the transmitter and receiver phase compensation FIFOs in the PCS.

(2) The refclk (refclk0 or refclk1) buffer is not powered down by this signal.

For more information, refer to *AN 558: Implementing Dynamic Reconfiguration in Arria II Devices*.

If none of the channels is instantiated in a transceiver block, the Quartus® II software automatically powers down the entire transceiver block.

Blocks Affected by Reset and Power-Down Signals

Table 4–2 lists the blocks that are affected by specific reset and power-down signals.

Table 4–2. Blocks Affected by Reset and Power-Down Signals for Arria II Devices (Part 1 of 2)

Transceiver Block	rx_digitalreset	rx_analogreset	tx_digitalreset	pll_powerdown	gxb_powerdown
CMU PLLs				✓	✓
Transmitter Phase Compensation FIFO			✓		✓
Byte Serializer			✓		✓
8B/10B Encoder			✓		✓
Serializer			✓		✓
Transmitter Buffer					✓
Transmitter XAUI State Machine			✓		✓
Receiver Buffer		✓ *(1)*			✓
Receiver CDR		✓			✓
Receiver Deserializer					✓
Receiver Word Aligner	✓				✓
Receiver Deskew FIFO	✓				✓
Receiver Clock Rate Compensation FIFO	✓				✓
Receiver 8B/10B Decoder	✓				✓
Receiver Byte Deserializer	✓				✓
Receiver Byte Ordering	✓				✓
Receiver Phase Compensation FIFO	✓				✓

Table 4–2. Blocks Affected by Reset and Power-Down Signals for Arria II Devices (Part 2 of 2)

Transceiver Block	rx_digitalreset	rx_analogreset	tx_digitalreset	pll_powerdown	gxb_powerdown
Receiver XAUI State Machine	✓				✓

Note to Table 4–2:

(1) RX signal detect in the RX buffer is disabled by rx_analogreset. However, RX signal detect is enabled when the busy signal is asserted.

The dynamic reconfiguration controller is always enabled when a receiver exists in a transceiver-based design even if the channel will not be reconfigured. The dynamic reconfiguration controller is used for receiver offset cancellation. Because of this, the busy signal from the dynamic reconfiguration controller must always be monitored when instantiating a receiver.

For more information about reset and offset cancellation, refer to the reset waveforms and the "Offset Cancellation" section of *AN 558: Implementing Dynamic Reconfiguration in Arria II Devices*.

Transceiver Reset Sequences

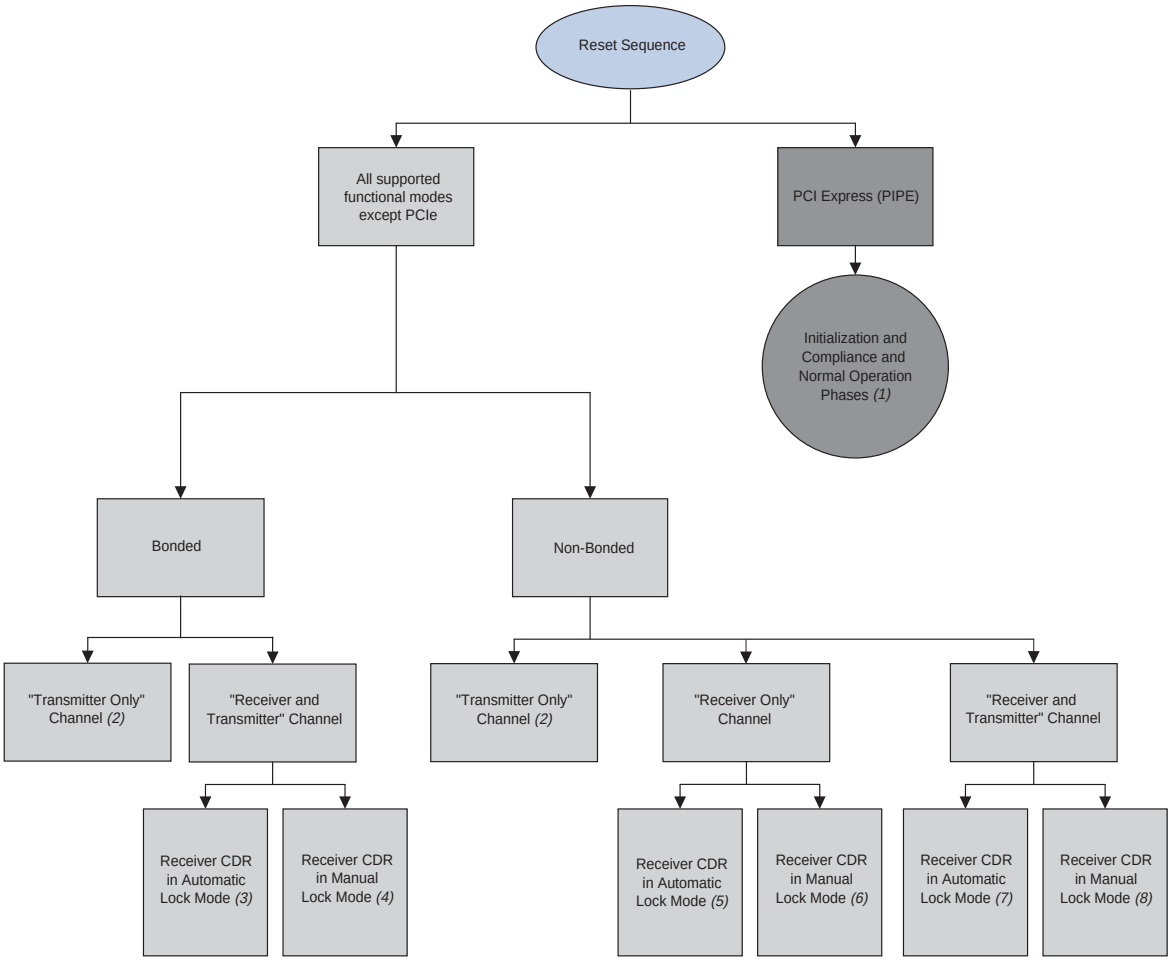
You can configure transceiver channels in Arria II GX and GZ devices in various configurations. In all functional modes except XAUI functional mode, transceiver channels can be either bonded or non-bonded. In XAUI functional mode, transceiver channels must be bonded. In PCI Express® (PIPE) (PCIe®) functional mode, transceiver channels can be either bonded or non-bonded and need to follow a specific reset sequence.

The two categories of reset sequences for the Arria II device family described in this chapter are:

- "All Supported Functional Modes Except PCIe Functional Mode" on page 4–6 describes the reset sequences in bonded and non-bonded configurations.
- "PCIe Functional Mode" on page 4–15 describes the reset sequence for the initialization/compliance phase and normal operation phase in PCIe functional modes.

The busy signal remains low for the first reconfig_clk clock cycle. It is then asserted from the second reconfig_clk clock cycle. Subsequent de-assertion of the busy signal indicates the completion of the offset cancellation process. This busy signal is required in transceiver reset sequences except for **Transmitter Only** channel configurations. Refer to the reset sequences shown in Figure 4–2 and the associated references listed in the notes.

Figure 4–2. Transceiver Reset Sequences Chart



Notes to Figure 4–2:

(1) Refer to the Timing Diagram in Figure 4–10 on page 4–15.
(2) Refer to the Timing Diagram in Figure 4–3 on page 4–7.
(3) Refer to the Timing Diagram in Figure 4–4 on page 4–8.
(4) Refer to the Timing Diagram in Figure 4–5 on page 4–9.
(5) Refer to the Timing Diagram in Figure 4–6 on page 4–11.
(6) Refer to the Timing Diagram in Figure 4–7 on page 4–12.
(7) Refer to the Timing Diagram in Figure 4–8 on page 4–13.
(8) Refer to the Timing Diagram in Figure 4–9 on page 4–14.

Altera strongly recommends adhering to these reset sequences for proper operation of the Arria II transceiver.

All Supported Functional Modes Except PCIe Functional Mode

This section describes the reset sequences for transceiver channels in bonded and non-bonded configurations. Timing diagrams of some typical configurations are shown to facilitate proper reset sequence implementation. In these functional modes, you can set the receiver CDR either in automatic lock or manual lock mode.

In manual lock mode, the receiver CDR locks to the reference clock (lock-to-reference) or the incoming serial data (lock-to-data), depending on the logic levels of the `rx_locktorefclk` and `rx_locktodata` signals. With the receiver CDR in manual lock mode, you can configure the transceiver channels in the Arria II GX or GZ device either in a non-bonded configuration or a bonded configuration. In a bonded configuration, such as XAUI mode, four channels are bonded together.

Table 4–3 lists the lock-to-reference (LTR) and lock-to-data (LTD) controller lock modes for the `rx_locktorefclk` and `rx_locktodata` signals.

Table 4–3. Lock-To-Reference and Lock-To-Data Modes for Arria II Devices

rx_locktorefclk	rx_locktodata	LTR/LTD Controller Lock Mode
1	0	Manual, LTR Mode
—	1	Manual, LTD Mode
0	0	Automatic Lock Mode

Bonded Channel Configuration

In a bonded channel configuration, you can reset all the bonded channels simultaneously. Examples of bonded channel configurations are XAUI, PCIe, and Basic x4 functional modes. In Basic x4 functional mode, you can bond **Transmitter Only** channels together.

In XAUI mode, the receiver and transmitter channels are bonded. Each of the receiver channels in this mode has its own output status signals, `rx_pll_locked` and `rx_freqlocked`. The timing of these signals is considered in the reset sequence.

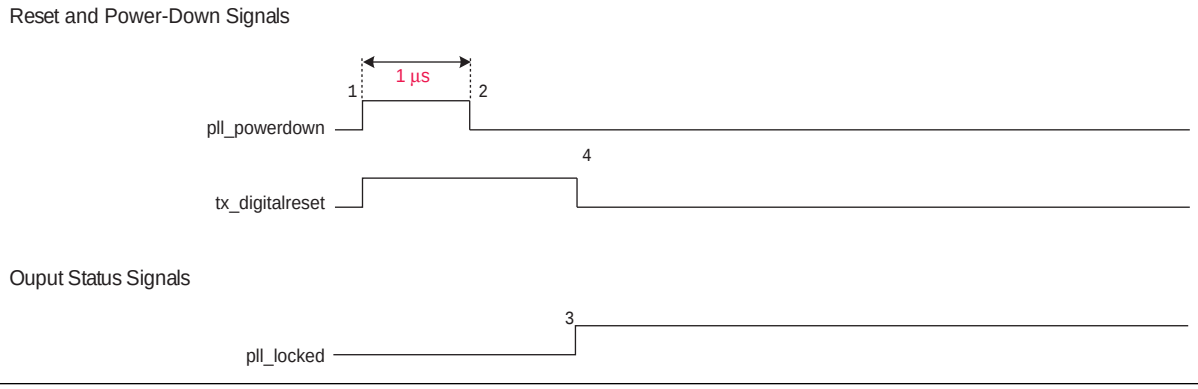
The following timing diagrams describe the reset and power-down sequences for bonded configurations under the following setups:

- **Transmitter Only** channel setup—applicable to Basic x4 functional mode
- **Receiver and Transmitter** channel setup—receiver CDR in automatic lock mode; applicable to XAUI functional mode
- **Receiver and Transmitter** channel setup—receiver CDR in manual lock mode; applicable to XAUI functional mode

Transmitter Only Channel

This configuration contains only a transmitter channel. If you create a **Transmitter Only** instance in the ALTGX MegaWizard Plug-In Manager in Basic x4 functional mode, use the reset sequence shown in Figure 4–3.

Figure 4–3. Sample Reset Sequence for Four Transmitter Only Channels



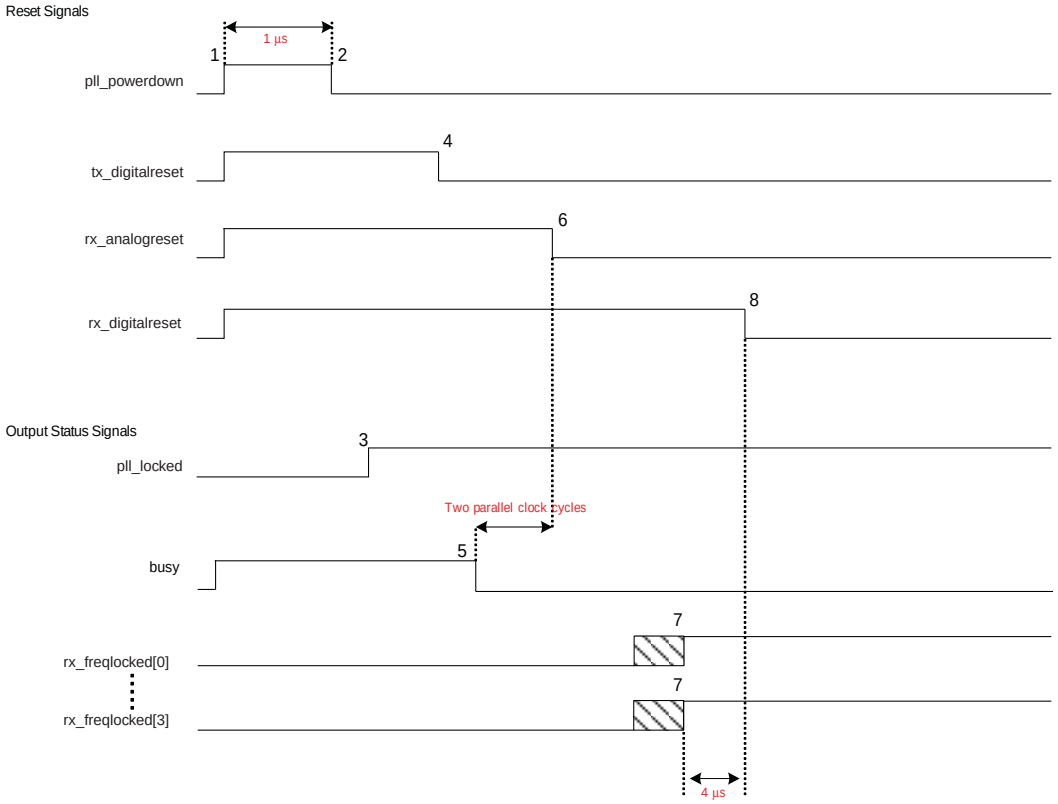
As shown in Figure 4–3, perform the following reset sequence steps for the **Transmitter Only** channel configuration:

1. After power up, assert `pll_powerdown` for a minimum period of 1 µs (the time between markers 1 and 2).
2. Keep the `tx_digitalreset` signal asserted during this time period. After you de-assert the `pll_powerdown` signal, the transmitter PLL starts locking to the transmitter input reference clock.
3. After the transmitter PLL locks, as indicated by the `pll_locked` signal going high (marker 3), de-assert the `tx_digitalreset` signal (marker 4). The transmitter is ready to transmit data.

Receiver and Transmitter Channel—Receiver CDR in Automatic Lock Mode

This configuration contains both a transmitter and receiver channel. For XAUI functional mode, with the receiver CDR in automatic lock mode, use the reset sequence shown in Figure 4–4.

Figure 4–4. Sample Reset Sequence for Four Receiver and Transmitter Channels—Receiver CDR in Automatic Lock Mode



As shown in Figure 4–4, perform the following reset sequence steps for the receiver CDR in automatic lock mode configuration:

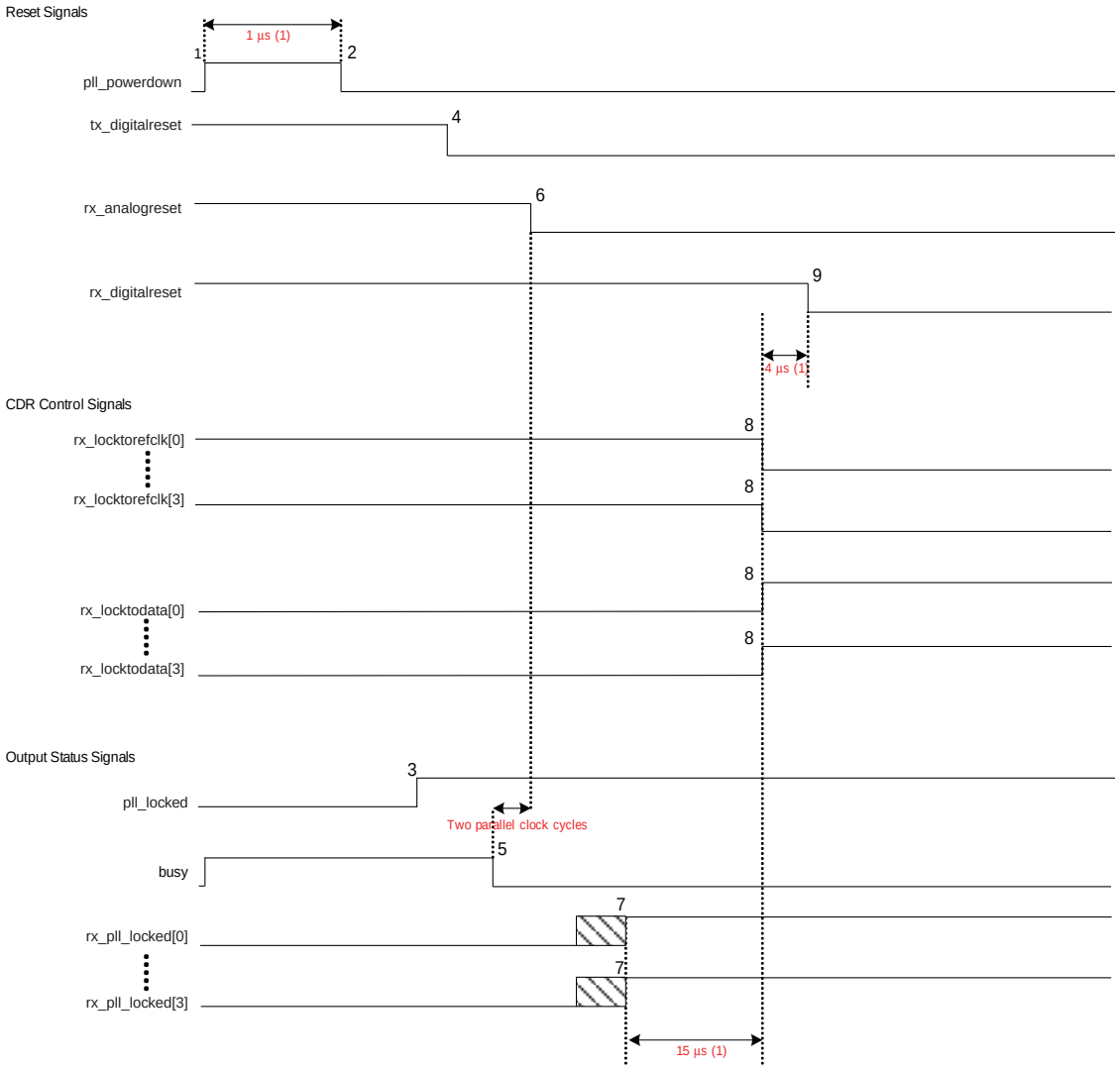
1. After power up, assert `pll_powerdown` for a minimum period of 1 µs (the time between markers 1 and 2).
2. Keep the `tx_digitalreset`, `rx_analogreset`, and `rx_digitalreset` signals asserted during this time period. After you de-assert the `pll_powerdown` signal, the transmitter PLL starts locking to the transmitter input reference clock.
3. After the transmitter PLL locks, as indicated by the `pll_locked` signal going high, de-assert the `tx_digitalreset` signal. At this point, the transmitter is ready for data traffic.
4. For the receiver operation, after de-assertion of the `busy` signal, wait for two parallel clock cycles to de-assert the `rx_analogreset` signal. After `rx_analogreset` is de-asserted, the receiver CDR of each channel starts locking to the receiver input reference clock.

5. Wait for the `rx_freqlocked` signal from each channel to go high. The `rx_freqlocked` signal of each channel may go high at different times (indicated by the slashed pattern at marker 7).
6. In a bonded channel group, when the `rx_freqlocked` signals of all the channels have gone high, from that point onwards, wait for at least 4 µs for the receiver parallel clock to be stable, then de-assert the `rx_digitalreset` signal (marker 8). At this point, all the receivers are ready for data traffic.

Receiver and Transmitter Channel—Receiver CDR in Manual Lock Mode

This configuration contains both a transmitter and receiver channel. For XAUI functional mode, with the receiver CDR in manual lock mode, use the reset sequence shown in Figure 4–5.

Figure 4–5. Sample Reset Sequence of Four Receiver and Transmitter Channels—Receiver CDR in Manual Lock Mode



Note to Figure 4–5:

(1) For the transceiver block power down duration, refer to the *Device Datasheet for Arria II Devices* chapter.

As shown in Figure 4–5, perform the following reset sequence steps for the receiver CDR in manual lock mode configuration:

1. After power up, assert pll_powerdown for a minimum period of 1 µs (the time between markers 1 and 2).
2. Keep the tx_digitalreset, rx_analogreset, rx_digitalreset, and rx_locktorefclk signals asserted and the rx_locktodata signal de-asserted during this time period. After you de-assert the pll_powerdown signal, the transmitter PLL starts locking to the transmitter input reference clock.
3. After the transmitter PLL locks, as indicated by the pll_locked signal going high (marker 3), de-assert the tx_digitalreset signal (marker 4). For the receiver operation, after de-assertion of the busy signal, wait for two parallel clock cycles to de-assert the rx_analogreset signal. After the rx_analogreset signal is de-asserted, the receiver CDR of each channel starts locking to the receiver input reference clock because rx_locktorefclk is asserted.
4. Wait for the rx_pll_locked signal from each channel to go high. The rx_pll_locked signal of each channel may go high at different times with respect to each other (indicated by the slashed pattern at the marker 7).
5. In a bonded channel group, when the last rx_pll_locked signal goes high, from that point onwards, wait at least 15 µs and then de-assert rx_locktorefclk and assert rx_locktodata (marker 8). At this point, the receiver CDR enters lock-to-data mode and the receiver PLL starts locking to the received data.
6. De-assert rx_digitalreset at least 4 µs (the time between markers 8 and 9) after asserting the rx_locktodata signal.

Non-Bonded Channel Configuration

In non-bonded channels, each channel in the ALTGX megafunction instance contains its own tx_digitalreset, rx_analogreset, rx_digitalreset, rx_pll_locked, and rx_freqlocked signals.

You can reset each channel independently. For example, if there are four non-bonded channels, the ALTGX MegaWizard Plug-In Manager provides five signals: tx_digitalreset, rx_analogreset, rx_digitalreset, rx_pll_locked, and rx_freqlocked.

The following timing diagrams describe the reset and power-down sequences for one channel in a non-bonded configuration, under five different setups:

- **Transmitter Only** channel setup
- **Receiver Only** channel setup—receiver CDR in automatic lock mode
- **Receiver Only** channel setup—receiver CDR in manual lock mode
- **Receiver and Transmitter** channel setup—receiver CDR in automatic lock mode
- **Receiver and Transmitter** channel setup—receiver CDR in manual lock mode

Follow the same reset sequence for all the other channels in the non-bonded configuration.

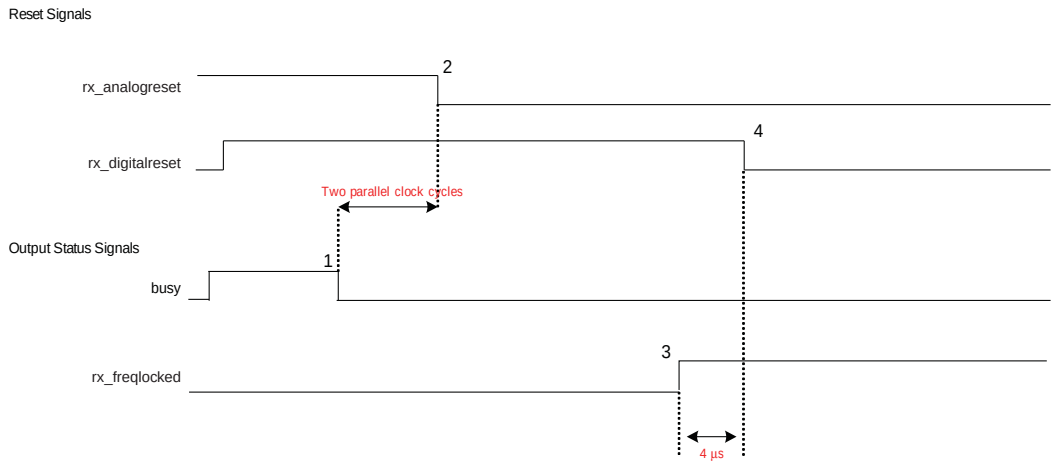
Transmitter Only Channel

This configuration contains only a transmitter channel. If you create a **Transmitter Only** instance in the ALTGX MegaWizard Plug-In Manager, use the same reset sequence as shown in Figure 4–2 on page 4–5.

Receiver Only Channel—Receiver CDR in Automatic Lock Mode

This configuration contains only a receiver channel. If you create a **Receiver Only** instance in the ALTGX MegaWizard Plug-In Manager with the receiver CDR in automatic lock mode, use the reset sequence shown in Figure 4–6.

Figure 4–6. Sample Reset Sequence of Receiver-Only Channel—Receiver CDR in Automatic Lock Mode



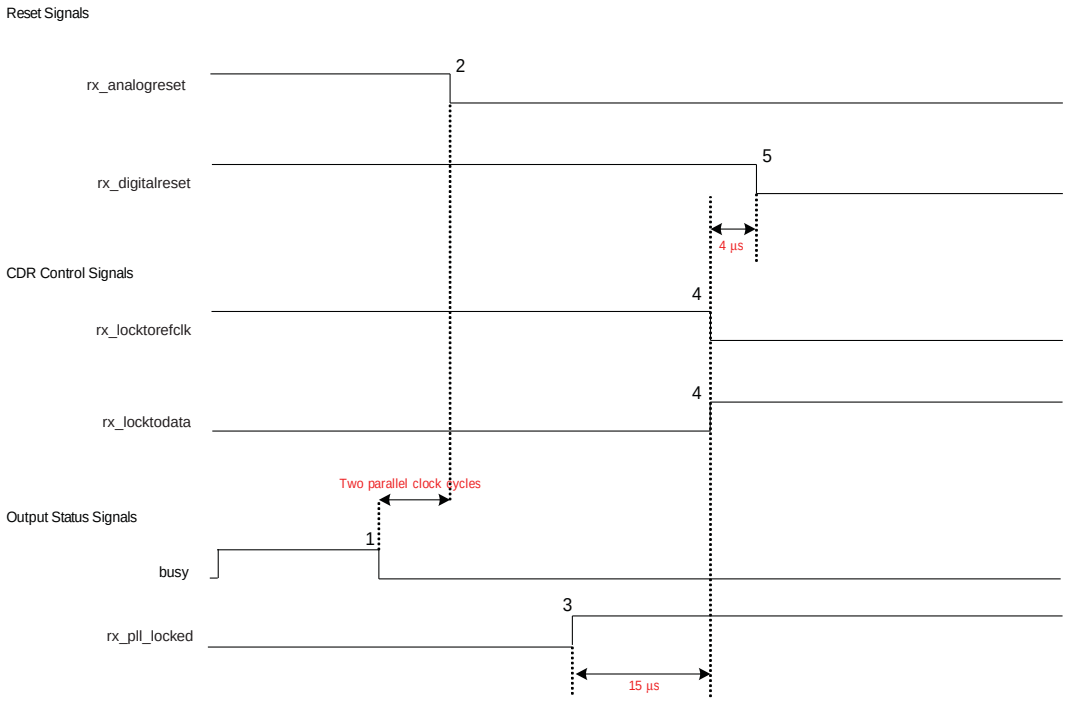
As shown in Figure 4–6, perform the following reset sequence steps for the receiver CDR in automatic lock mode configuration:

1. After power up, wait for the `busy` signal to be de-asserted (marker 1).
2. De-assert the `rx_analogreset` signal (marker 2).
3. Keep the `rx_digitalreset` signal asserted during this time period. After you de-assert the `rx_analogreset` signal, the receiver PLL starts locking to the receiver input reference clock.
4. Wait for the `rx_freqlocked` signal to go high (marker 3).
5. After `rx_freqlocked` goes high, wait at least 4 µs and then de-assert the `rx_digitalreset` signal (marker 4). At this point, the receiver is ready to receive data.

Receiver Only Channel—Receiver CDR in Manual Lock Mode

This configuration contains only a receiver channel. If you create a **Receiver Only** instance in the ALTGX MegaWizard Plug-In Manager with the receiver CDR in manual lock mode, use the reset sequence shown in Figure 4–7.

Figure 4–7. Sample Reset Sequence of Receiver-Only Channel—Receiver CDR in Manual Lock Mode



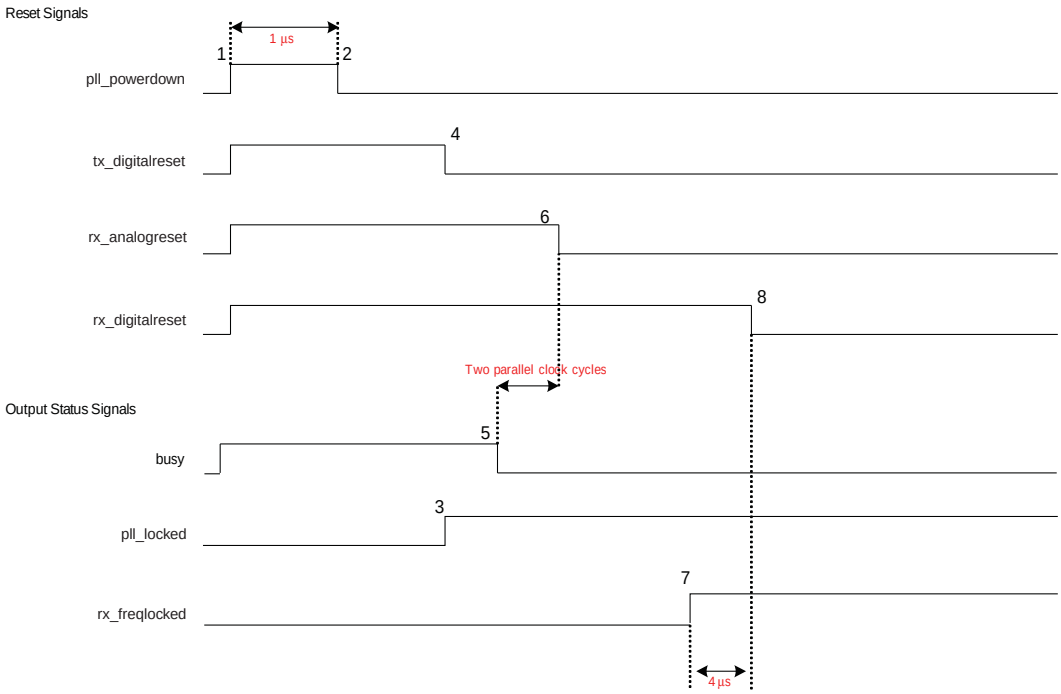
As shown in Figure 4–7, perform the following reset sequence steps for the receiver CDR in manual lock mode:

1. After power up, assert `rx_analogreset` for a minimum period of two parallel clock cycles (the time between markers 1 and 2).
2. Keep the `rx_digitalreset` and `rx_locktorefclk` signals asserted and the `rx_locktodata` signal de-asserted during this time period.
3. After de-assertion of the `busy` signal, de-assert the `rx_analogreset` signal, after which the receiver CDR starts locking to the receiver input reference clock because the `rx_locktorefclk` signal is asserted.
4. Wait at least 15 µs (the time between markers 3 and 4) after the `rx_pll_locked` signal goes high, and then de-assert the `rx_locktorefclk` signal. At the same time, assert the `rx_locktodata` signal (marker 4). At this point, the receiver CDR enters lock-to-data mode and the receiver PLL starts locking to the received data.
5. De-assert `rx_digitalreset` at least 4 µs (the time between markers 4 and 5) after asserting the `rx_locktodata` signal.

Receiver and Transmitter Channel—Receiver CDR in Automatic Lock Mode

This configuration contains both a transmitter and receiver channel. If you create a **Receiver and Transmitter** instance in the ALTGX MegaWizard Plug-In Manager with the receiver CDR in automatic lock mode, use the reset sequence shown in Figure 4–8.

Figure 4–8. Sample Reset Sequence of Receiver and Transmitter Channel—Receiver CDR in Automatic Lock Mode



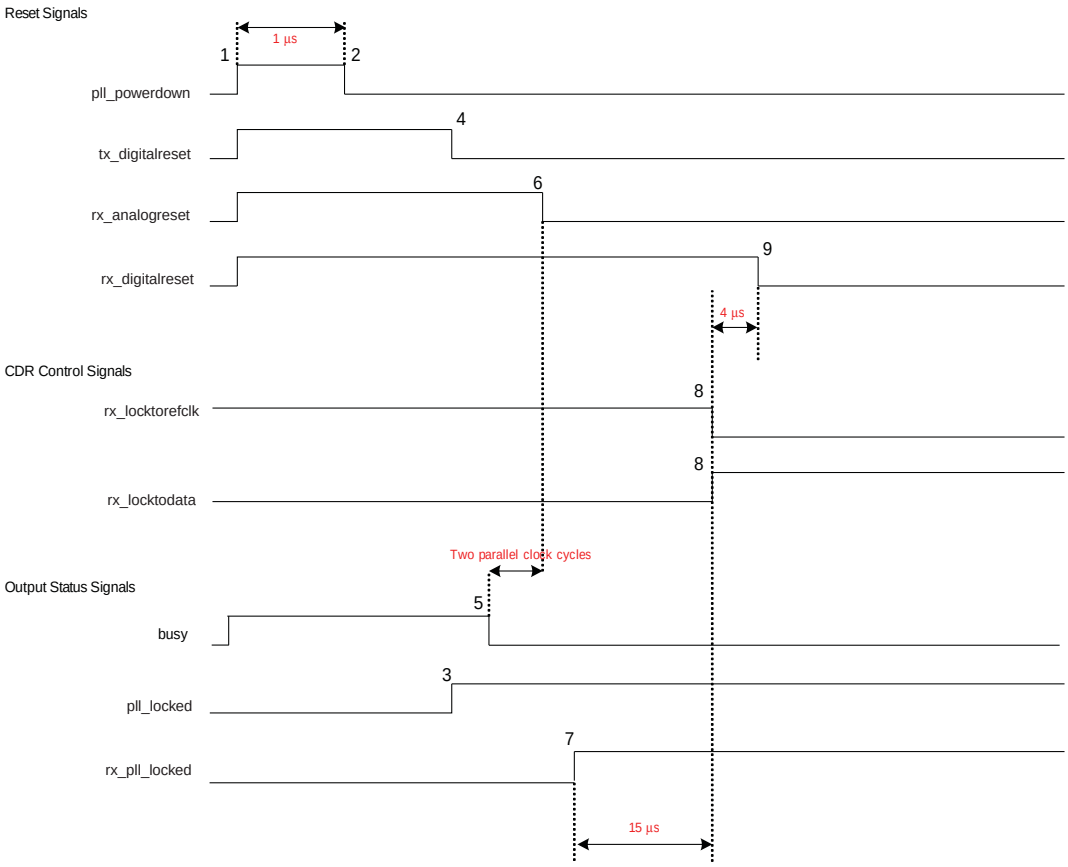
As shown in Figure 4–8, perform the following reset sequence steps for the receiver CDR in automatic lock mode:

1. After power up, assert `pll_powerdown` for a minimum period of 1 µs (the time between markers 1 and 2).
2. Keep the `tx_digitalreset`, `rx_analogreset`, and `rx_digitalreset` signals asserted during this time period. After you de-assert the `pll_powerdown` signal, the transmitter PLL starts locking to the transmitter input reference clock.
3. After the transmitter PLL locks, as indicated by the `pll_locked` signal going high (marker 3), de-assert `tx_digitalreset`. For receiver operation, wait for the `busy` signal to be de-asserted, after which `rx_analogreset` is de-asserted. After you de-assert `rx_analogreset`, the receiver CDR starts locking to the receiver input reference clock.
4. Wait for the `rx_freqlocked` signal to go high (marker 7).
5. After the `rx_freqlocked` signal goes high, wait at least 4 µs and then de-assert the `rx_digitalreset` signal (marker 8). The transmitter and receiver are ready for data traffic.

Receiver and Transmitter Channel—Receiver CDR in Manual Lock Mode

This configuration contains both a transmitter and receiver channel. If you create a **Receiver and Transmitter** instance in the ALTGX MegaWizard Plug-In Manager with the receiver CDR in manual lock mode, use the reset sequence shown in Figure 4–9.

Figure 4–9. Sample Reset Sequence of Receiver and Transmitter Channel—Receiver CDR in Manual Lock Mode



As shown in Figure 4–9, perform the following reset sequence steps for the receiver in manual lock mode:

1. After power up, assert `pll_powerdown` for a minimum period of 1 µs (the time between markers 1 and 2).
2. Keep the `tx_digitalreset`, `rx_analogreset`, `rx_digitalreset`, and `rx_locktorefclk` signals asserted and the `rx_locktodata` signal de-asserted during this time period. After you de-assert the `pll_powerdown` signal, the transmitter PLL starts locking to the transmitter input reference clock.
3. After the transmitter PLL locks, as indicated by the `pll_locked` signal going high (marker 3), de-assert `tx_digitalreset`. For receiver operation, wait for the `busy` signal to be de-asserted. At this point, `rx_analogreset` is de-asserted. After `rx_analogreset` is de-asserted, the receiver CDR starts locking to the receiver input reference clock because `rx_locktorefclk` is asserted.

4. Wait for at least 15 µs (the time between markers 7 and 8) after the `rx_pll_locked` signal goes high, then de-assert the `rx_locktorefclk` signal. At the same time, assert the `rx_locktodata` signal (marker 8). At this point, the receiver CDR enters lock-to-data mode and the receiver CDR starts locking to the received data.
5. De-assert `rx_digitalreset` at least 4 µs (the time between markers 8 and 9) after asserting the `rx_locktodata` signal.

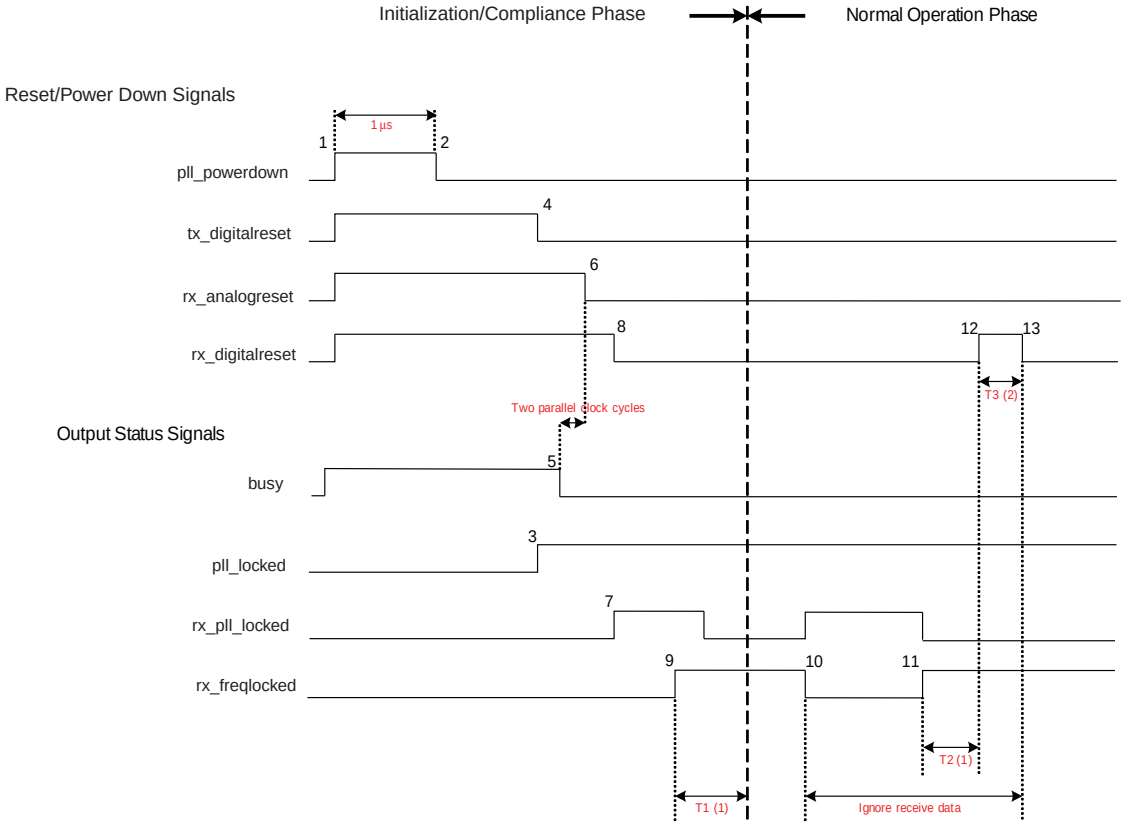
PCIe Functional Mode

You can configure PCIe functional mode with or without the receiver clock rate compensation FIFO in the Arria II device family. The reset sequence remains the same of whether or not you use the receiver clock rate compensation FIFO.

PCIe Reset Sequence

PCIe protocol consists of the initialization/compliance phase and normal operation phase. The reset sequences for these two phases are based on the timing diagram shown in Figure 4–10.

Figure 4–10. Reset Sequence of PCIe Functional Mode



Notes to Figure 4–10:

(1) The minimum T1 and T2 period is 4 µs.
(2) The minimum T3 period is two parallel clock cycles.

PCIe Initialization/Compliance Phase

After the device is powered up, a PCIe-compliant device goes through the compliance phase during initialization. The rx_digitalreset signal must be de-asserted during this compliance phase to achieve transitions on the pipephydonestatus signal, as expected by the link layer.

The rx_digitalreset signal is de-asserted based on the assertion of the rx_freqlocked signal.

During the initialization/compliance phase, do not use the rx_freqlocked signal to trigger a de-assertion of the rx_digitalreset signal. Instead, perform the following reset sequence as shown in Figure 4–10:

1. After power up, assert pll_powerdown for a minimum period of 1 µs (the time between markers 1 and 2). Keep the tx_digitalreset, rx_analogreset, and rx_digitalreset signals asserted during this time period. After you de-assert the pll_powerdown signal, the transmitter PLL starts locking to the transmitter input reference clock.
2. After the transmitter PLL locks, as indicated by the pll_locked signal going high (marker 3), de-assert tx_digitalreset. For receiver operation, wait for the busy signal to be de-asserted and for rx_analogreset to be de-asserted. After rx_analogreset is de-asserted, the receiver CDR starts locking to the receiver input reference clock.
3. When the receiver CDR locks to the input reference clock, as indicated by the rx_pll_locked signal going high at marker 7, de-assert the rx_digitalreset signal (marker 8). After de-asserting rx_digitalreset, the pipephydonestatus signal transitions from the transceiver channel to indicate the status to the link layer. Depending on its status, pipephydonestatus helps with the continuation of the compliance phase. After successful completion of this phase, the device enters into the normal operation phase.

PCIe Normal Phase

For the normal operation phase, perform the following reset sequence, as shown in Figure 4–10:

1. After completion of the Initialization/Compliance phase, when the rx_freqlocked signal is de-asserted, (marker 10), wait for the rx_pll_locked signal assertion signifying the lock-to-reference clock.
2. Wait for the rx_freqlocked signal to go high again. In this phase, the received data is valid (not electrical idle) and the receiver CDR locks to the incoming data.
3. Proceed with the reset sequence after assertion of the rx_freqlocked signal.
4. After the rx_freqlocked signal goes high, wait for at least 4 µs before asserting rx_digitalreset (marker 12) for two parallel receive clock cycles so that the receiver phase compensation FIFO is initialized.

Data from the transceiver block is not valid from the time the rx_freqlocked signal goes low (marker 10) to the time rx_digitalreset is de-asserted (marker 13). The PLD logic ignores the data during this period (between markers 10 and 13).

You can configure the Arria II GX or GZ device in x1, x2, x4, and x8 PIPE lane configurations. The reset sequence described in "PCIe Reset Sequence" on page 4–15 applies to all these multi-lane configurations.

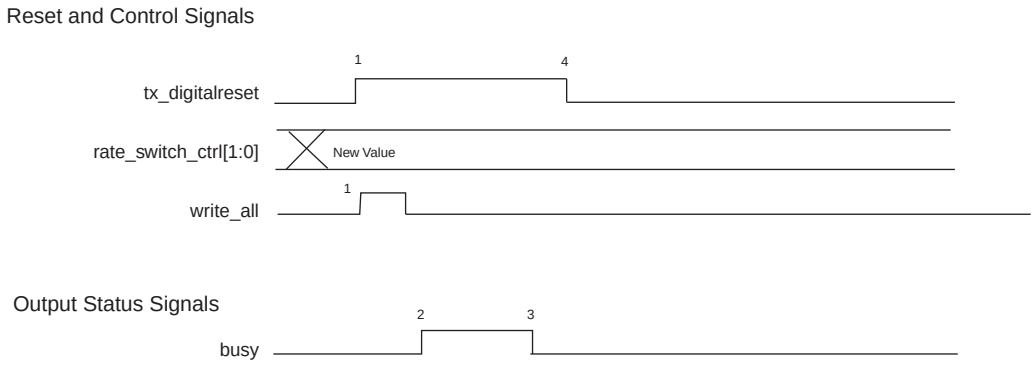
Dynamic Reconfiguration Reset Sequences

When using dynamic reconfiguration in data rate divisions in TX or Channel and TX CMU PLL select/reconfig modes, use the following reset sequences.

Reset Sequence with Data Rate Division in the TX Option

Use the example reset sequence shown in Figure 4–11 when you are using the dynamic reconfiguration controller to change the data rate of the transceiver channel. In this example, dynamic reconfiguration is used to dynamically reconfigure the data rate of the transceiver channel configured in Basic x1 mode with the receiver CDR in automatic lock mode.

Figure 4–11. Reset Sequence in Basic x1 Mode with the Receiver CDR in Automatic Lock Mode (TX Option)



As shown in Figure 4–11, perform the following reset procedure when using the dynamic reconfiguration controller to change the configuration of the transmitter channel:

1. After power up and properly establishing that the transmitter is operating correctly, write the new value for the data rate in the appropriate register (in this example, `rate_switch_ctrl[1:0]`) and subsequently assert the `write_all` signal (marker 1) to initiate the dynamic reconfiguration.

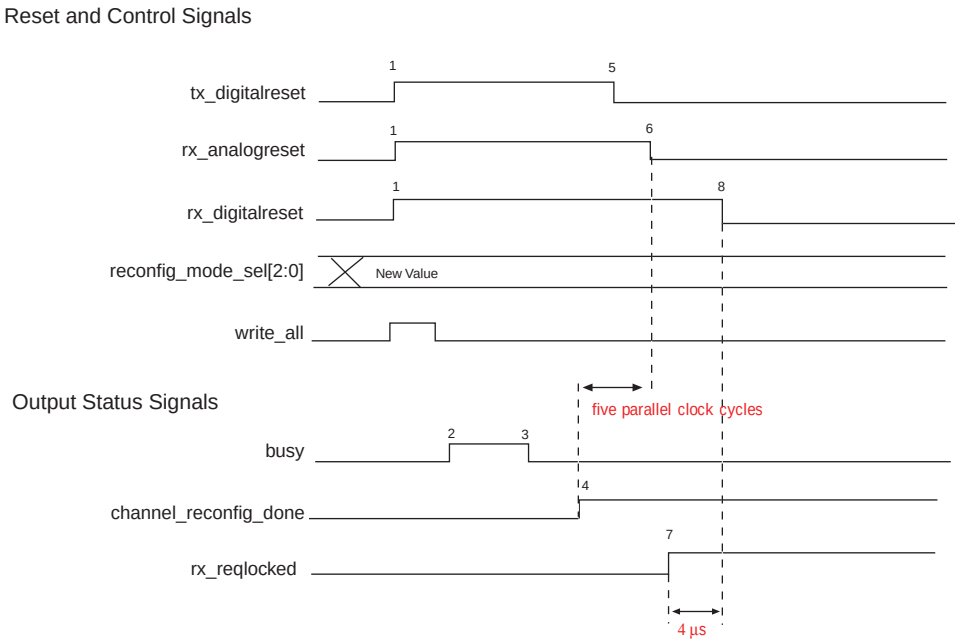
   For more information, refer to *AN 558: Implementing Dynamic Reconfiguration in Arria II Devices*.

2. Assert the `tx_digitalreset` signal.
3. As soon as `write_all` is asserted, the dynamic reconfiguration controller starts to execute its operation, as indicated by the assertion of the `busy` signal (marker 2).
4. After the completion of dynamic reconfiguration, the `busy` signal is de-asserted (marker 3).
5. Finally, `tx_digitalreset` can be de-asserted to continue with the transmitter operation (marker 4).

Reset Sequence with the Channel and TX PLL Select/Reconfig Option

Use the example reset sequence shown in Figure 4–12 when you are using the dynamic reconfiguration controller to change the TX PLL settings of the transceiver channel. In this example, the dynamic reconfiguration is used to dynamically reconfigure the data rate of the transceiver channel configured in Basic x1 mode with the receiver CDR in automatic lock mode.

Figure 4–12. Reset Sequence in Basic x1 Mode with Receiver CDR in Automatic Lock Mode (Channel and TX PLL select/reconfig Option)



As shown in Figure 4–12, perform the following reset procedure when using the dynamic reconfiguration controller to change the configuration of the transceiver channel:

1. After power up and establishing that the transceiver is operating correctly, write the new value in the appropriate registers (including `reconfig_mode_sel[2:0]`) and subsequently assert the `write_all` signal (marker 1) to initiate the dynamic reconfiguration.

   For more information, refer to *AN 558: Implementing Dynamic Reconfiguration in Arria II Devices*.

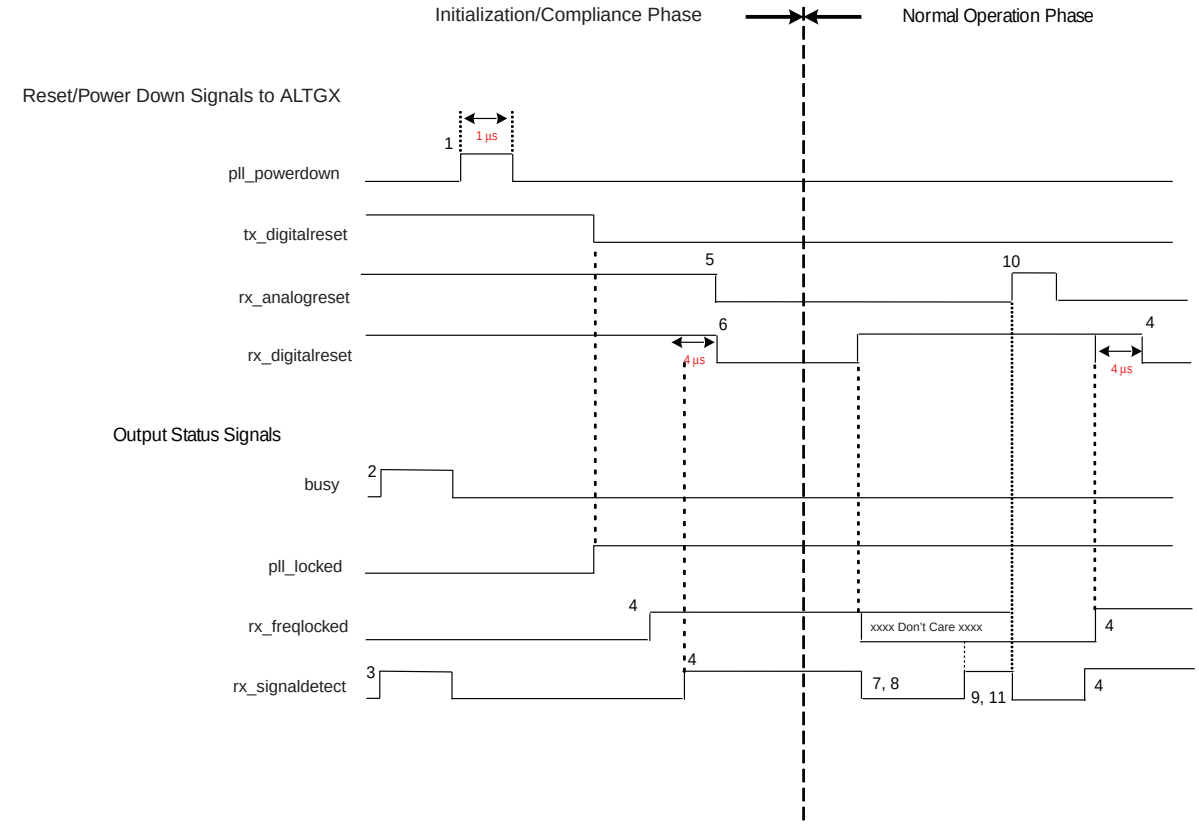
2. Assert the `tx_digitalreset`, `rx_analogreset`, and `rx_digitalreset` signals.
3. As soon as `write_all` is asserted, the dynamic reconfiguration controller starts to execute its operation, as indicated by the assertion of the `busy` signal (marker 2).
4. Wait for the assertion of the `channel_reconfig_done` signal (marker 4), which indicates the completion of dynamic reconfiguration in this mode.

5. After assertion of the `channel_reconfig_done` signal, de-assert `tx_digitalreset` (marker 5) and wait for at least five parallel clock cycles to de-assert the `rx_analogreset` signal (marker 6).
6. Finally, wait for the `rx_freqlocked` signal to go high. After `rx_freqlocked` goes high (marker 7), wait for 4 µs to de-assert the `rx_digitalreset` signal (marker 8). At this point, the receiver is ready for data traffic.

Hot Socketing Reset Sequence

Use the example hot socketing reset sequence shown in Figure 4–13 when you are using hot socketing.

Figure 4–13. Reset Sequence for Hot Socketing



As shown in Figure 4–13, perform the following reset procedure when using hot socketing to change the configuration of the transceiver channel:

1. After power up, assert `pll_powerdown` for a minimum period of 1 µs.
2. When `busy` is asserted, the RX signal detect block is enabled.
3. When `busy` is asserted, `rx_signaldetect` is asserted if the RX input is valid or de-asseted if the RX input is invalid.
4. If `rx_freqlocked` is asserted when `rx_signaldetect` is still low, keep `rx_digitalreset` asserted.

5. Wait until rx_signaldetect is asserted, then re-assert rx_analogreset for at least two parallel clock cycle.
6. De-assert rx_digitalreset when rx_freqlocked and rx_signaldetect are asserted for ≥ 4 µs.
7. If the RX input is invalid, rx_signaldetect is de-asserted.
8. When rx_signaldetect de-asserts, assert rx_digitalreset to avoid RX from receiving corrupted data.
9. Monitor rx_signaldetect until rx_signaldetect is asserted.
10. If the RX input is not floating or the far end TX input is not in electrical idle, rx_signaldetect is asseted when the RX input is valid.
11. When rx_signaldetect is asserted, assert the rx_analogreset for two parallel clock cycles.
12. The rx_freqlocked and rx_signaldetect is de-asserted when rx_analogreset is asserted.

For other functional modes that do not have the rx_signaldetect output port but require hot socketing, you have several options available to determine the link status. You can logically AND together any combinations of these options and use them alternatively as a loss-of-link status indicator.

- Option 1: Create or use the Receiver Monitor Block to observe the rx word alignment status and monitor the received data at the upper layer. Repeat step 4 of the "Hot Socketing Reset Sequence" on page 4–19 if the receiver losses word alignment synchronization or detects an error in the received data.
- Option 2: Implement a Digital Loss of Signal detector by enabling the **Run Length Violation** option in the Word Aligner to detect bit transitions in a predetermined sliding window length.
- Option 3: Implement a parts per million (PPM) detector in the device core to detect loss-of-link status.
- Option 4: Use the Receiver Phase Compensation FIFO overflow/underflow status port to detect loss-of-link status.

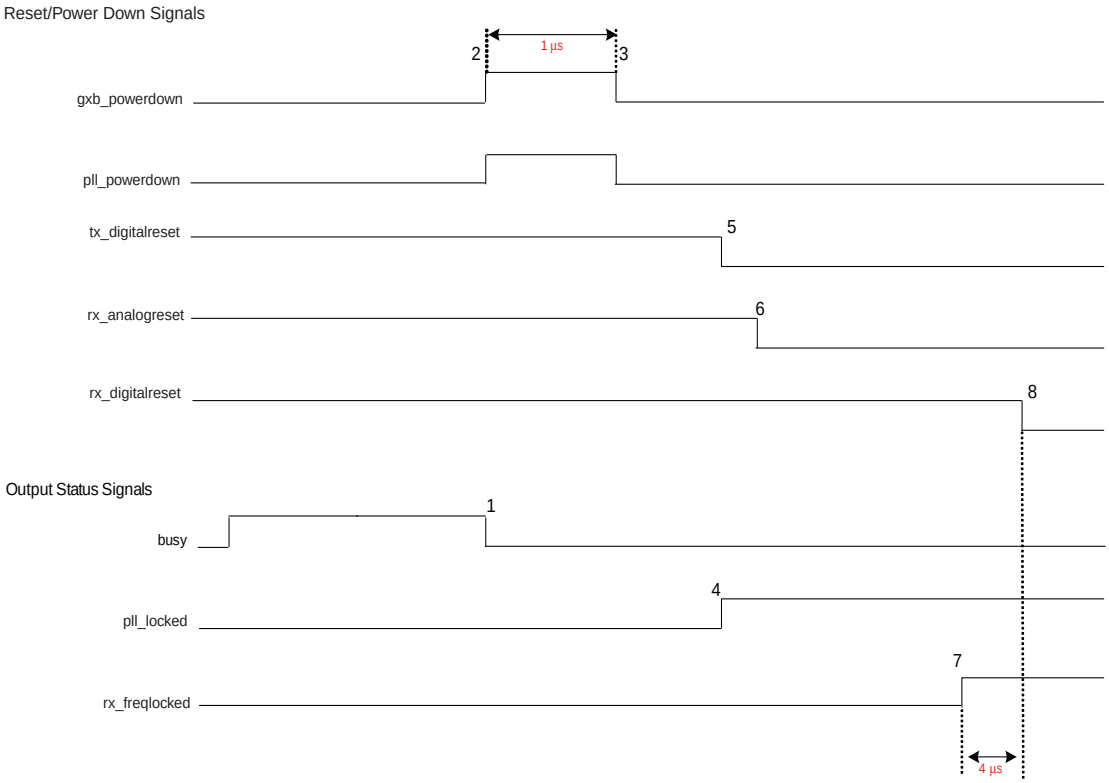
Power Down

The Quartus II software automatically selects the power down channel feature, which takes effect when you configure the Arria II GX or GZ device. All unused transceiver channels and blocks are powered down to reduce overall power consumption.

The gxb_powerdown signal is an optional transceiver block signal. It powers down all the blocks in the transceiver block. The minimum pulse width for this signal is 1 µs.

After power up, if you use the gxb_powerdown signal, wait for de-assertion of the busy signal, then assert the gxb_powerdown signal for a minimum of 1 µs. To finish, follow the sequence shown in Figure 4–14.

Figure 4–14. Sample Reset Sequence of Four Receiver and Transmitter Channels—Receiver CDR in Automatic Lock Mode with Optional gxb_powerdown Signal



Simulation Requirements

The following are simulation requirements:

- The gxb_powerdown port is optional. In simulation, if the gxb_powerdown port is not instantiated, you must assert the tx_digitalreset, rx_digitalreset, and rx_analogreset signals appropriately for correct simulation behavior.
- If the gxb_powerdown port is instantiated, and the other reset signals are not used, you must assert the gxb_powerdown signal for at least one parallel clock cycle for correct simulation behavior.
- You can de-assert the rx_digitalreset signal immediately after the rx_freqlocked signal goes high to reduce the simulation run time. It is not necessary to wait 4 µs (as suggested in the actual reset sequence).
- The busy signal is de-asserted after approximately 20 parallel reconfig_clk clock cycles in order to reduce the simulation run time. For silicon behavior in the hardware, follow the reset sequences described in this chapter.
- In PCIe mode simulation, you must assert the tx_forceelecidle signal for at least one parallel clock cycle before transmitting normal data for correct simulation behavior.

Document Revision History

Table 4–4 lists the revision history for this chapter.

Table 4–4. Document Revision History

Date	Version	Changes
June 2011	3.1	■ Updated Table 4–1 and Table 4–2. ■ Added the "Hot Socketing Reset Sequence" section. ■ Minor text edits.
December 2010	3.0	■ Updated to add Arria II GZ information. ■ Minor text edits.
July 2010	2.0	■ Updated Figure 4–4, Figure 4–5, and Figure 4–12. ■ updated the "Blocks Affected by Reset and Power-Down Signals" section. ■ Minor text edits.
March 2009	1.1	Added the "Dynamic Reconfiguration Reset Sequences" section.
February 2009	1.0	Initial release.