Implementing SATA and SAS Protocols in Altera Devices

AN-635-1.0 Application Note

This application note describes how to implement the Serial Advanced Technology Attachment (SATA) and Serial Attached SCSI (SAS) protocols with Altera® transceivers in the Arria® II, HardCopy® IV, and Stratix® IV devices. You can create your proprietary SATA and SAS interface designs using FPGA logic to interface with the transceiver configurations described in this document.

This document contains the following topics:

- "SATA and SAS Protocols Overview" on page 1
- "Transceiver Support for SATA and SAS Protocols" on page 2
- "Transceiver Configuration for SATA and SAS" on page 4
- "Setting up the Transceiver for SATA and SAS Applications" on page 5
- "Reconfiguring Transceiver for SATA and SAS Speed Negotiation" on page 8
- "Reconfiguring SATA and SAS Data Rates Using the ALTGX_RECONFIG Controller" on page 10
- "Reset Sequence for SATA and SAS During Initialization, Hot-plugged, and After Dynamic Reconfiguration" on page 13
- "Timing Closure Involving Speed Negotiation" on page 18

SATA and SAS Protocols Overview

SATA and SAS are two serial connections protocols between hosts and peripherals in desktops, servers, and other applications. The protocols are similar in terms of data rate and signal requirements but are targeted for slightly different applications.

SATA is an evolution of the parallel ATA interface that was developed for use as an interconnect for desktop PCs, servers, and enterprise systems to connect a host system to peripheral devices such as hard disk drives and optical storage drives. SATA systems are designed to operate in half-duplex mode—communication can only take place in one direction at a time. The physical data rates for SATA are 1.5 Gbps, 3.0 Gbps, and 6.0 Gbps.

SAS protocol is used as interconnect between disk drives and host controllers mainly in server systems. SAS is designed to be backwards compatible with SATA systems while offering more features—far more capacity, easier scalability and expansion, and enhanced security.

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

The SAS protocol is designed to operate in full-duplex mode—data can be transmitted and received to and from the drive simultaneously. The protocol supports connections with up to 8 m cables and can support the use of expanders that allow for connections of multiple SAS drives to a single host port. The physical data rates for SAS are 3.0 Gbps and 6.0 Gbps.

Transceiver Support for SATA and SAS Protocols

Altera transceiver devices support the SATA and SAS standards—each version of the standards primarily addresses the speed of a single data link. Table 1 lists the Altera transceiver FPGA and HardCopy ASIC SATA and SAS support.

Table 1. SATA and SAS Support

Device	SATA			SAS	
	SATA 1.0 (1.5 Gbps)	SATA 2.0 (3.0 Gbps)	SATA 3.0 (6.0 Gbps)	SAS 1.0 (3.0 Gbps)	SAS 2.0 (6.0 Gbps)
Stratix IV	✓	✓	✓	✓	✓
Arria II GX	✓	✓	✓ *(1)*	✓	✓ *(1)*
Arria II GZ	✓	✓	✓ *(2)*	✓	✓ *(2)*
HardCopy IV GX	✓	✓	✓	✓	✓

Notes to Table 1:

(1) Supported only in the –I3 speed grade of Arria II GX devices.

(2) Supported only in the –C3 and –I3 speed grades of the Arria II GZ devices (EP2AGZ225, EP2AGZ300, and EP2AGZ350).

This document describes the recommended transceiver configuration for implementing SATA and SAS protocols to fulfill PHY requirements.

The Arria II, HardCopy IV, and Stratix IV devices are compliant with SATA and SAS PHY requirements—natively supported by Altera transceivers. You can easily implement protocol coding, out-of-band signaling, and speed negotiation with the 8B/10B block, receiver signal detect, transmitter electrical idle, and dynamic reconfiguration features.

The following sections describe the Altera transceiver devices hardware compliance to SATA and SAS specifications:

- "Out-of-Band Signaling for SATA and SAS" on page 2
- "Automatic Speed Negotiation for SATA and SAS" on page 3
- "Spread-Spectrum Input Clock Tracking" on page 4

Out-of-Band Signaling for SATA and SAS

SATA and SAS protocols both support the out-of-band signaling scheme. Out-of-band signaling is used for the following functions:

- Establish communication between a host and drive to identify the type of drive used in the system
- Identify the maximum operating data rate of the host and drive

An out-of-band signal is a tri-level signal that contains a pattern of idle and burst signals. Out-of-band signaling is used to signal specific actions during conditions where the receiving interface is inactive or in low power state mode. The out-of-band signals comprise of COMRESET, COMINIT/COMWAKE, and COMSAS. You can use Altera devices to generate and detect out-of-band sequences through the transmitter electrical idle and receiver signal detect features.

While generating the idle signals between burst signals, you can instantiate the tx_forceelecidle port at the transmitter to send an idle period based on the out-of-band signals specifications. The tx_forceelecidle signal is used to tri-state the transmitter buffer.

To enable this particular signal, turn on **Create 'tx_forceelecidle' input port** at the **Ports/Calibration** page of the ALTGX megafunction. For more information, refer to "Transceiver Configuration Using ALTGX MegaWizard Plug-In Manager" on page 7.

You can use the receiver signal detect feature to detect the presence of each COMRESET, COMINIT/COMWAKE, and COMSAS signal during out-of-band sequence. The threshold level for these signals varies based on the SATA and SAS usage model—the i, m, and x variants of SATA (Gen1—1.5 Gbps, Gen2—3.0 Gbps, and Gen3—6.0 Gbps) and SAS (Gen1—3.0 Gbps, and Gen2—6.0 Gbps).

To instantiate this port in the ALTGX megafunction, turn on **Create 'rx_signaldetect' port to indicate data input signal detection** at the **Ports/Calibration** setting page. For more information, refer to "Transceiver Configuration Using ALTGX MegaWizard Plug-In Manager" on page 7.

There are two signal detect threshold levels available for selection in the ALTGX megafunction. You can select the specific signal detect threshold based on the SATA and SAS usage model. Use the following signal detect threshold settings:

- Gen1i/m, Gen2i/m, Gen3i—Signal detect threshold = 2
- Gen1x, Gen2x—Signal detect threshold = 6

For more information on the SATA and SAS usage models such as for the i, m, and x variants, refer to the protocols specifications.

You can set the receiver input rx_signaldetect threshold using the Quartus II software version 9.1 and later.

Automatic Speed Negotiation for SATA and SAS

SATA and SAS speed negotiation takes place between the host and device to determine the maximum data rate that the host and device can support. This negotiation is a PHY requirement in the SATA and SAS specifications.

The automatic speed negotiation process sends out a steady stream of patterns from both nodes on the link. The process observes the out-of-band signaling pattern to modify the data rate and negotiate a common speed.

The Altera transceiver dynamic reconfiguration feature allows you to reconfigure to different data rates during speed negotiation. To choose a suitable reconfiguration method for this application, refer to "Reconfiguring Transceiver for SATA and SAS Speed Negotiation" on page 8, and followed by the recommended reset sequence, "Reset Sequence for SATA and SAS During Initialization, Hot-plugged, and After Dynamic Reconfiguration" on page 13.

Spread-Spectrum Input Clock Tracking

Spread-spectrum clocking (SSC) is required to reduce electromagnetic interference (EMI) emission. Altera transceiver devices cannot internally generate spread-spectrum clocks. For spread-spectrum clock generation, you must use an external clock source with SSC generation.

However, Altera receiver CDR PLLs are designed to accept a spread-spectrum input with modulated frequencies. Typically, the CDR PLLs are able to track the SSC if you have configured the CDR PLLs to either medium or high bandwidth setting and the frequencies are within the input-jitter tolerance specifications.

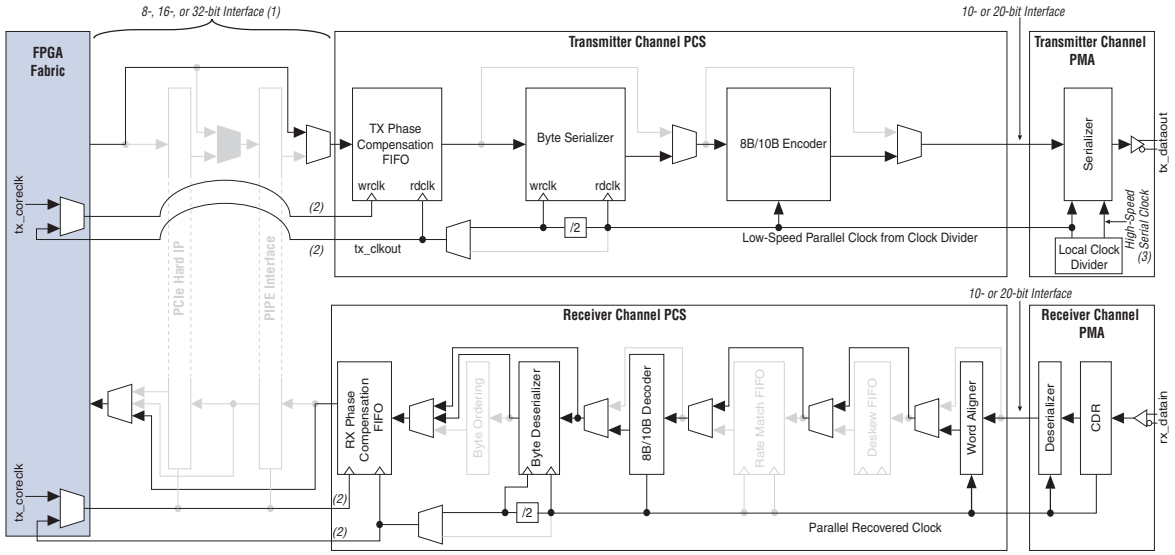
For more information about the input-jitter tolerance specification, refer to the following documents:

- The *DC and Switching Characteristics for Stratix IV Devices* chapter in the *Stratix IV Device Handbook*.
- The *DC and Switching Characteristics of HardCopy IV Devices* chapter in the *HardCopy IV Device Handbook*.
- The *Device Datasheet for Arria II Devices* chapter in the *Arria II Device Handbook*.

Transceiver Configuration for SATA and SAS

Figure 1 shows the transceiver configuration using Basic Functional Mode to run in SATA and SAS protocols.

Figure 1. Transceiver Datapath for SATA Configuration



Notes to Figure 1:

(1) The 32-bit interface is supported only in the Stratix IV, HardCopy IV GX, and Arria II GZ devices.

(2) These clocks runs at 37.5 MHz, 75 MHz, or 150 MHz, depending on the SATA and SAS data rate, and the SERDES factor (single- and double-width modes). For more information, refer to "Select the Supported Channel Width Settings for SATA and SAS" on page 6.

(3) The high-speed serial clock runs at 750 MHz (1.5 Gbps), 1500 MHz (3 Gbps), and 3000 MHz (6 Gbps), depending on the SATA and SAS data rates.

Setting up the Transceiver for SATA and SAS Applications

The following sections describe how to set up the transceiver for SATA and SAS:

- "Using Recommended PLL configuration for SATA and SAS" on page 5
- "Choosing an Input Reference Clock" on page 5
- "Select the Supported Channel Width Settings for SATA and SAS" on page 6
- "Transceiver Configuration Using ALTGX MegaWizard Plug-In Manager" on page 7

Using Recommended PLL configuration for SATA and SAS

You must select the most suitable PLL configuration settings to operate across SATA and SAS data rates of 1.5 Gbps, 3.0 Gbps, and 6.0 Gbps. Table 2 lists the recommended PLL configuration for SATA and SAS protocols.

Table 2. Recommended PLL Configuration for SATA and SAS Protocols

Protocol Data Rate	Base Data Rate (Mbps) *(1)*	Effective Data Rate (Mbps) *(2)*	TX PLL Bandwidth	CDR PLL Bandwidth
SATA 1.0	6000 *(3)*	1500	Automatic	Medium or High
	3000			
	1500			
SATA 2.0/SAS 1.0	6000 *(3)*	3000		
	3000			
SATA 3.0/SAS 2.0	6000 *(3)*	6000 *(3)*		

Notes to Table 2:

(1) Base data rate is the frequency at the output of the transmit PLL VCO. For information about the usage of this setting, refer to "Transceiver Configuration Using ALTGX MegaWizard Plug-In Manager" on page 7.

(2) Effective data rate is the serial data rate at the channel after the channel local divider. For information about the usage of this setting, refer to "Transceiver Configuration Using ALTGX MegaWizard Plug-In Manager" on page 7.

(3) The 6000 Mbps effective data rate is supported only in the Stratix IV, HardCopy IV, Arria II GX (–I3 speed grade), and Arria II GZ (–I3 and –C3 speed grades) devices.

Choosing an Input Reference Clock

If you create your designs using the Arria II, HardCopy IV, and Stratix IV GX/GT (except Stratix IV GX 230 ES) devices, Altera recommends an input reference clock frequency of 150 MHz for all supported data rates. However, for designs using the Stratix IV GX 230 ES device, follow the restrictions listed in Table 3 when you select an input reference clock.

The Stratix IV GX 230 ES devices have restricted input reference clock frequency settings for both transmit and receive PLLs. If you choose an input reference clock frequency and data rate that use the restricted 'M' counter settings, the Quartus II compiler exits with an error.

Table 3. Restricted Input Reference Clock for the Stratix IV GX 230 ES Devices

Device	Base Data Rate *(1)*	Not Allowed for 230 ES	
		M=16	M=20
Stratix IV GX 230 ES	3000 Mbps	93.75 MHz	75 MHz
	6000 Mbps	187.5 MHz	150 MHz

Note to Table 3:

(1) Base data rate is the frequency at the output of the PLL VCO.

Select the Supported Channel Width Settings for SATA and SAS

Table 4 lists the supported line rate, channel width, and interface clock rates settings for the Arria II, HardCopy IV, and Stratix IV devices. Select the most suitable settings for your transceiver configuration.

Table 4. Supported Line Rate, Channel Width, and Interface Clock Rates Settings

Protocol	Line Rate (Mbps)	Channel Width (bits) *(1)*	Interface Clock Rates (MHz)	
			Stratix IV and HardCopy IV	Arria II
SATA 1.0	1500	8	150	150
		16	75	75
		32	37.5	37.5 *(2)*
SATA 2.0/SAS 1.0	3000	16	150	150
		32	75	75 *(2)*
SATA 3.0/SAS 2.0	6000	16	300	300 *(3)*
		32	150	150 *(2)*

Notes to Figure 5:

(1) The 8-, 16-, and 32-bit channel widths are supported with the 8B/10B block. The 16-bit channel width is supported with the double-width mode or the byte serializer/deserializer (SERDES) block. The 32-bit channel width is supported with both the double-width mode and the byte SERDES block.

(2) Applicable only for the Arria II GZ and only at the –I3 and –C3 speed grades.

(3) Applicable only for the Arria II GX (–I3 speed grade) and Arria II GZ (–I3 and –C3 speed grades).

Transceiver Configuration Using ALTGX MegaWizard Plug-In Manager

Navigate through the **MegaWizard™ Plug-In Manager** and specify the necessary options and settings. Table 5 lists the specific values to configure the transceiver channel to operate in SATA and SAS.

Table 5. Specific ALTGX Megafunction Options for Transceiver Channel Operation in SATA and SAS (Part 1 of 2)

Option	Settings
Which megafunction would you like to customize	Expand **I/O** and select **ALTGX**.
Which protocol will you be using?	Select **Basic**.
What is the operation mode?	Based on your design, select **Receiver and Transmitter**, **Transmitter Only**, or **Receiver Only**.
What is the number of channels?	Select **1**.
What is the channel width?	Select the channel width. *(1)*
What is the effective data rate?	Type the desired channel line rate. *(2)*
What is the input clock frequency?	Select **150.0** MHz.
The base data rate is	Type the base data rate. *(2)*
What is the Tx PLL bandwidth mode?	Select **Auto**.
What is Rx CDR bandwidth mode?	Select **High** or **Medium**.
Create 'gxb_powerdown' port to powerdown the Transceiver block	Turn on. This setting is optional.
Create 'rx_analogreset' port for the analog portion of the receiver	Turn on.
Create 'rx_digitalreset' port for the digital portion of the receiver	Turn on.
Create 'tx_digitalreset' port for the digital portion of the transmitter	Turn on.
Create 'rx_locktorefclk' port to lock the Rx CDR to the reference clock	Turn on.
Create 'rx_locktodata' port to lock the Rx CDR to the received data	Turn on.
Create 'rx_signaldetect' port to indicate data input signal detection *(3)*	Turn on.
Create 'tx_forceelecidle' input port *(4)*	Turn on.
What is the signal detect threshold?	Select the following values: ■ Gen1i/m, Gen2i/m, and Gen3i—**2** ■ Gen1x and Gen2x—**6**
Settings on the **TX Analog** page	Set the V_{OD} according to the SATA and SAS specifications. You can refer to the relevant device literature for the details. You can use dynamic reconfiguration of the analog controls to modify the transmitter V_{OD}.
Analog control(VOD, Pre-emphasis, Manual Equalization and EyeQ)	Turn on.

Table 5. Specific ALTGX Megafunction Options for Transceiver Channel Operation in SATA and SAS (Part 2 of 2)

Option	Settings
Enable Channel and Transmitter PLL reconfiguration	Turn on to enable switching the channel data rate during speed negotiation.
Enable 8b10b decoder/encoder	Turn on to use the `rx_signaldetect` port.
Create 'rx_ctrldetect' to indicate 8b10b decoder has detected a control code *(5)*	Turn on.
Word Aligner options group	Select **Use manual word alignment mode**.
What is the word alignment pattern length?	Specify **10**.
What is the word alignment pattern?	Type `0101111100` (10-bit K28.5– alignment pattern). The word aligner detects both disparities of this pattern.

Notes to Table 5:

(1) Set the channel width based on Table 4 on page 6.

(2) If the channel requires speed negotiation capability, set the highest line rate intended for the device. Negotiation to lower line rate is supported using the dynamic reconfiguration feature. For example, if you intend to run the channel at 1500 Mbps and plan to reconfigure it to 6000 Mbps, set the base data rate to 6000 Mbps and the effective data rate to 1500 Mbps.

(3) You can use this port to detect the out-of-band signaling and the presence of data at the receiver buffer.

(4) This port is used to send idle signals at the common mode voltage during the out-of-band signaling sequence. For more information about achieving out-of-band signaling, refer to "Out-of-Band Signaling for SATA and SAS" on page 2.

(5) This option is available if you turn on the **Enable 8b10b decoder/encoder** option.

SATA and SAS specification requires clock compensation of +350/–5350 ppm. The hard-coded Rate Match FIFO in the physical coding sublayer (PCS) can only compensate for up to ±300 ppm between the transmitter and receiver clocks. You can implement the rate match FIFO in the FPGA fabric using the 8-bit word and control bit output from the receiver PCS in single-width mode and 16-bit word and 2-bit control output from the receiver PCS in double-width mode.

CAUTION

You should disable the rate matcher operation in the **Rate match/Byte order** page for the receiver datapath in the ALTGX megafunction if not already disabled for your configuration. You should also disable **Byte Ordering Block** if not already disabled for your configuration. Implement the byte ordering block in the FPGA fabric because byte ordering occurs after clock compensation (rate matcher).

Reconfiguring Transceiver for SATA and SAS Speed Negotiation

You can use the dynamic reconfiguration feature to reconfigure the SATA and SAS transceiver channels to achieve speed negotiation when the host and device are establishing the communication to identify the maximum operating data rate of the drive.

The following sections describe the steps to set up the reconfiguration controller for speed negotiation.You can use the ALTGX_RECONFIG MegaWizard Plug-In Manager to meet this requirement.

Setup for SATA and SAS Reconfiguration during Automatic Speed Negotiation

Before you perform dynamic reconfiguration between SATA and SAS data rates, you must first generate the Memory Initialization Files (**.mif**). Follow these steps:

1. Turn on the **Generate GXB Reconfig MIF** option:
   a. On the **Assignments** menu, click **Settings**.
   b. In the **Category** list, click **Fitter Settings**. Then, under **Fitter Settings**, click **More Settings**.
   c. In the **More Fitter Settings** window, in the **Name** box, select **Generate GXB Reconfig MIF**. Then, in the **Setting** box, select **On**.
2. Create an **ALTGX** instance for the SATA Gen1 (1.5 Gbps) configuration. Compile a design and create a **.mif** and name it **SATA_MIF_Gen1.mif**.
3. Create an **ALTGX** instance for the SATA Gen2 (3 Gbps) configuration. Compile a design and create a **.mif** and name it **SATA_MIF_Gen2.mif**.
4. When the **.mif** files are ready, you can configure the device based on the ALTGX_RECONFIG MegaWizard Plug-in Manager design flow by using one of the methods explained in the following sections.

Design Flow Using the ALTGX_RECONFIG MegaWizard Plug-in Manager

You can use this method to reconfigure the SATA and SAS channels during speed negotiation. The following example uses the Stratix IV GX device configured in SATA. The same settings can be applied to the Arria II and HardCopy IV devices for the same supported data rates.

Navigate through the **MegaWizard™ Plug-In Manager** and specify the necessary options and settings. Table 6 lists the specific values to instantiate a transceiver reconfiguration controller.

Table 6. Specific ALTGX_RECONFIG Megafunction Options to Instantiate the Transceiver Reconfiguration Controller (Part 1 of 2)

Option	Settings
Which megafunction would you like to customize	Expand **I/O** and select **ALTGX_RECONFIG**.
What is the number of channels controlled by the reconfig controller?	Specify the number of channels.

Table 6. Specific ALTGX_RECONFIG Megafunction Options to Instantiate the Transceiver Reconfiguration Controller (Part 2 of 2)

Option	Settings
What are the features to be reconfigured by the reconfig controller?	Turn on the following options: ■ To dynamically reconfigure the transmitter V_{OD}, pre-emphasis, and receiver equalization, turn on **Analog controls**. ■ To reconfigure only the transmitter datapath data rate across SATA data rates, turn on **Data rate division in TX**. ■ To reconfigure the TX and RX CDR PLL across SATA data rates, turn on **Channel and TX PLL select/reconfig**. This also allows you to reconfigure channel parameters such as the channel data width.
Settings on the **Analog controls** page	Select the analog parameters that you want to control, or leave the options at default settings.
Use 'reconfig_address_en'	Turn on.
Use 'reset_reconfig_address'	Turn on.
Use 'reconfig_address' port to input address from MIF in reduced MIF reconfiguration	Turn on.
Enable self recovery *(1)*	Turn on.
Use 'rate_switch_out' port to read out the current data rate division	Turn on.
Use 'rx_tx_duplex_sel' port to enable RX only, TX only or duplex reconfiguration	Turn on to reconfigure the transmitter and receiver channels independently of each other.

Note to Table 5:

(1) The Quartus II software can recover from an incomplete operation that has timed out and drives the error port high.

For more information about the port lists, refer to the *Dynamic Reconfiguration in Stratix IV Devices* chapter of the *Stratix IV Device Handbook*, the *HardCopy IV GX Dynamic Reconfiguration* chapter of the *HardCopy IV Device Handbook*, and *AN 558: Implementing Dynamic Reconfiguration in Arria II Devices*.

Reconfiguring SATA and SAS Data Rates Using the ALTGX_RECONFIG Controller

You can use one of the following methods to achieve the 54.6 µs fast reconfiguration time required by the SATA and SAS specification for both your transmitter and receiver channels. The 54.6 µs timing requirement includes the configuration time, and the reset and relocked sequence.

To comply with the 54.6 µs timing requirement, Altera recommends the following reconfiguration methods.

Reconfiguring the Transmitter Channel Using the Data Rate Division in TX Mode

This method uses the **Data Rate Division in TX** mode and it affects only the channel divider settings.

This mode allows dynamic reconfiguration of the transmitter local divider settings to 1, 2, or 4. The transmitter channel data rate is reconfigured based on the local divider settings.

For example, SATA Gen1, Gen2, and Gen3 data rates are multiples—or divisions—of two or four. You can reconfigure the data rates by using the **Data Rate Division in TX** mode. Set the base data rate to the highest SATA line rate (6.0 Gbps, in this example) supported by the device and set the TX local divider to one of the following data rate division values:

- To reconfigure the transmitter to the Gen1 data rate, set the data rate division to 4.
- To reconfigure the transmitter to the Gen2 data rate, set the data rate division to 2.
- To reconfigure the transmitter to the Gen3 data rate, set the data rate division to 1.

For more information about the **Data Rate Division in TX** mode, refer to the *Dynamic Reconfiguration in Stratix IV Devices* chapter of the *Stratix IV Device Handbook*, the *HardCopy IV GX Dynamic Reconfiguration* chapter of the *HardCopy IV Device Handbook*, and *AN 558: Implementing Dynamic Reconfiguration in Arria II Devices*.

Reconfiguring the Receiver Channel Using the Reduced .mif Reconfiguration Mode

This method uses the **reduced .mif reconfiguration** mode and it affects only the modified settings of the receiver channel.

Using this method, only the intended changed settings for your transceiver channel are updated, which significantly reduces reconfiguration time. To select the receiver channel that you want to reconfigure, enable the `rx_tx_duplex_sel` port and set its 2'b01 value. You can use a reconfiguration clock frequency of 50 MHz for this operation.

The **reduced .mif reconfiguration** mode is supported by the Quartus II software version 9.1 and later only for the Arria II GX, HardCopy IV, and Stratix IV devices.

How To Perform the Reduced .mif Reconfiguration for SATA and SAS

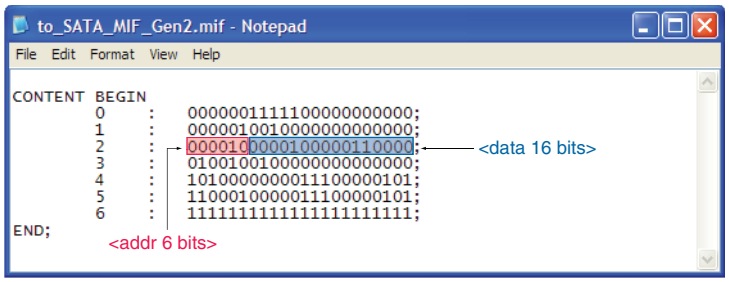
This section describes the method to use the **reduced .mif reconfiguration** mode to perform speed negotiation for SATA and SAS.

This mode is available only for the **.mif**-based transceiver channel reconfiguration mode. This mode is an optional feature that allows faster reconfiguration and shorter simulation time. If you want to make minor changes to the transceiver channel, you can use this mode because it involves only a few words in the **.mif**.

In the following example, assume that the only difference between two **.mif** files is the word address 32. Instead of loading the entire **.mif**, you can use the **altgx_diffmifgen.exe** executable to generate a new **.mif**. This new **.mif** only has the modified word (word address 32). As shown in Figure 2, the new **.mif** is 22 bits wide, compared to the 16 bits wide regular **.mif**. There are six bits of address in addition to the 16 bits of data:

<addr 6 bits> <data 16 bits>

Figure 2. Example of a Reduced .mif Showing the 6 Bits of Address and 16 Bits of Data



Before you generate the reduced **.mif** files, configure these options in the **Channel and TX PLL Reconfiguration** page of the ALTGX_RECONFIG MegaWizard Plug-In Manager:

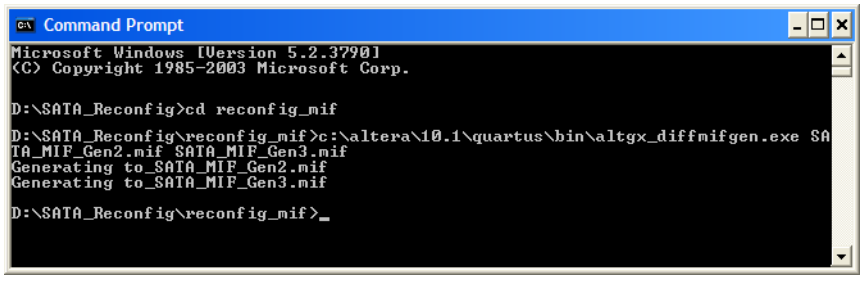
- Turn on the `reconfig_address` port to input the address bits from the **.mif**.
- Connect the `reconfig_data[15:0]` port to the 16 least significant bit (LSB) of the word from the new **.mif**.
- Connect the `reconfig_address[5:0]` port to the 6 most significant bit (MSB) of the word from the new **.mif**.

To generate the reduced **.mif** files, follow these steps:

1. Find the installed folder of the Quartus II software that contains the **altgx_diffmifgen.exe** executable (for example, **C:\altera\10.1\quartus\bin**) and take note of this directory path.
2. Launch the Command Prompt and navigate to the **reconfig_mif** folder of your project directory. The **reconfig_mif** folder contains the **.mif** files that you generated earlier—the **SATA_MIF_Gen2.mif** and **SATA_MIF_Gen3.mif** files. Then, type the following command:

   <path to Quartus II bin folder>`\altgx_diffmifgen.exe SATA_MIF_Gen2.mif SATA_MIF_Gen3.mif ↵`

Figure 3. Example of Generating the Reduced .mif File in the Project Directory



From the input files (**SATA_MIF_Gen2.mif** and **SATA_MIF_Gen3.mif**), the **altgx_diffmifgen.exe** executable generates two **.mif** files:

- **to_SATA_MIF_Gen2.mif**—Use this file to reconfigure the transceiver to the Gen2 data rate.
- **to_SATA_MIF_Gen3.mif**—Use this file to reconfigure the transceiver to the Gen3 data rate.

You can use the described methods to perform the reduced **.mif** reconfiguration for your transmitter and duplex channels. Manipulate the `rx_tx_duplex_sel` port to select either the transmitter, receiver, or duplex to suit your application.

For more information about channel and TX PLL reconfiguration, **.mif** generation, and **.mif**-based design flow, refer to the *Stratix IV Dynamic Reconfiguration* chapter in the *Stratix IV Device Handbook*, the *HardCopy IV GX Dynamic Reconfiguration* chapter in the *HardCopy IV Device Handbook*, and *AN 558: Implementing Dynamic Reconfiguration in Arria II Devices*.

Reset Sequence for SATA and SAS During Initialization, Hot-plugged, and After Dynamic Reconfiguration

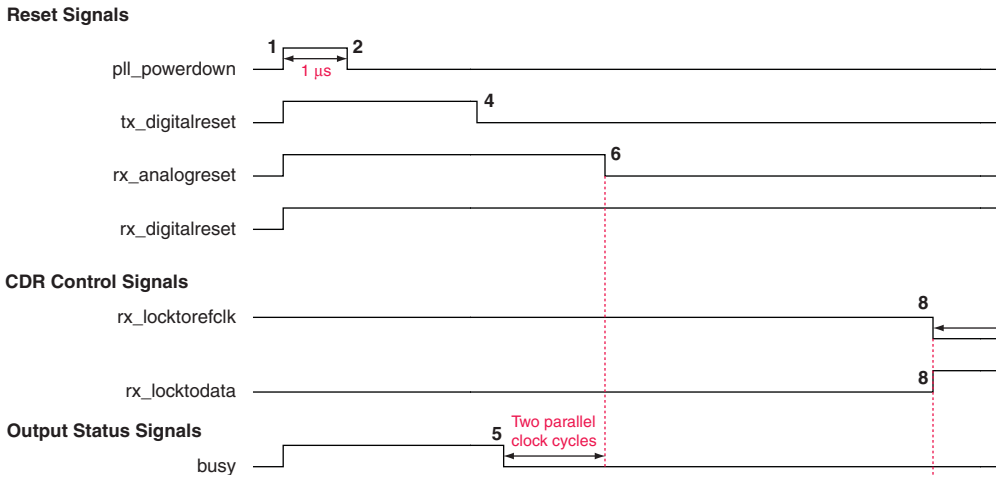
To ensure the correct operation of the transceiver in your SATA and SAS applications, instantiate the ports and control the reset signals according to the reset sequences recommended in this section for the following situations:

- During the link initialization
- During a hot-plugged condition
- After the completion of the dynamic reconfiguration process

Initializing SATA and SAS Channel with Reset Sequence

Figure 4 shows the recommended reset sequence to control the reset controller signals after a power-up condition—for a non-bonded configuration mode with the receiver CDR configured in manual lock mode.

Figure 4. Reset Sequence to Initialize the SATA and SAS Links in Manual Lock Mode



Note to Figure 4:

(1) The `rx_signaldetect` signal toggles when the RX buffer receives the out-of-band (OOB) signals that are sent in the burst and idle modes. After the OOB signaling, continuous data is sent to the receiver buffer and the `rx_signaldetect` signal latches high—indicating that valid data is being received.

As shown in Figure 4, follow these reset steps:

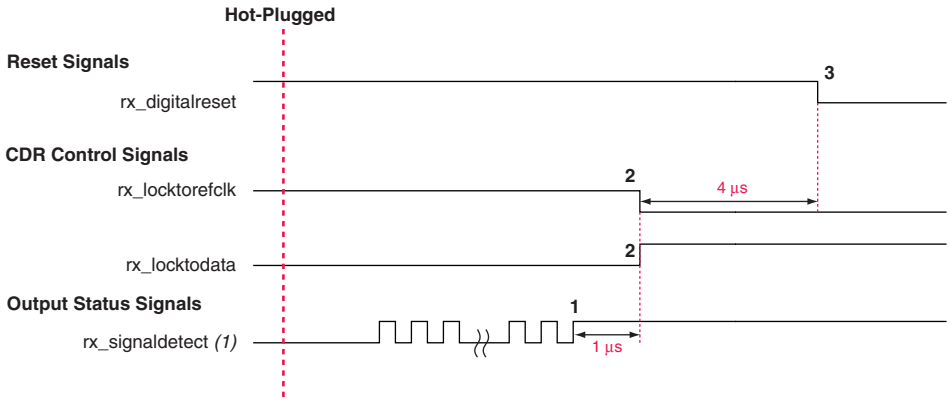
1. After power up, assert the `pll_powerdown` signal for at least 1 µs (markers 1 and 2). Keep the `tx_digitalreset`, `rx_analogreset`, `rx_digitalreset`, and `rx_locktorefclk` signals asserted, and the `rx_locktodata` signal deasserted during this period.
2. After you deassert the `pll_powerdown` signal, the transmitter PLL starts locking to the transmitter input reference clock.
3. After the `pll_locked` signal goes high (marker 3), deassert the `tx_digitalreset` signal. For the receiver operation, wait for the `busy` signal to be deasserted before you deassert the `rx_analogreset` signal. The receiver CDR starts locking to the receiver input reference clock.
4. When the first `rx_signaldetect` signal goes high, start to monitor this signal. Once the `rx_signaltdetect` latches high (marker 7)—indicating that valid data are present at the receiver buffer—wait for at least 1 µs, and then deassert the `rx_locktorefclk` signal and assert the `rx_locktodata` signal. At this point, the receiver CDR enters the lock-to-data mode and starts locking to the received data.
5. Deassert the `rx_digitalreset` signal for at least 4 µs (markers 8 and 9) after asserting the `rx_locktodata` signal. At this point, the receiver is ready for data traffic.

Reset Sequence During Hot-Plugged Condition

During a hot-unplugged and hot-plugged condition, you can monitor the signal detect status signal (rx_signaldetect)—indicating the link status—to perform a proper reset sequence when data is reintroduced to the SATA and SAS receiver link.

Figure 5 shows the recommended timing diagram for using the rx_signaldetect signal to control the reset sequence if your receiver CDR is configured in the manual lock mode.

Figure 5. Reset Sequence After a Hot-Plugged Condition in Manual Lock Mode



Note to Figure 5:

(1) The rx_signaldetect signal toggles when the RX buffer receives the out-of-band (OOB) signals that are sent in the burst and idle modes. After the OOB signaling, continuous data is sent to the receiver buffer and the rx_signaldetect signal latches high—indicating that valid data is being received.

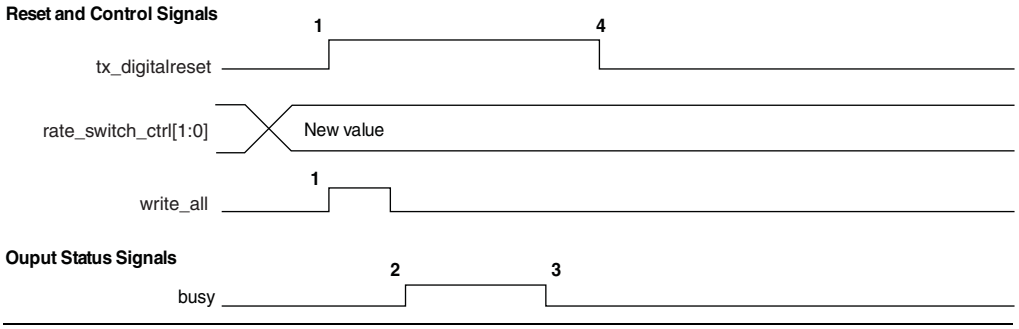
As shown in Figure 5, follow these reset steps:

1. To handle a hot-plugged reset, monitor the rx_signaldetect status signal and wait for the signal to latch high (marker 1), indicating reception of continuos valid data at the receiver.
2. Once the rx_signaldetect signal latches high, wait for at least 1 µs and then deassert the rx_locktorefclk signal and assert the rx_locktodata signal. The CDR switches from the lock-to-reference mode to the lock-to-data mode.
3. Wait 4 µs after asserting the rx_locktodata signal (marker 2) and then deassert the rx_digitalreset signal (marker 3). The receiver starts receiving valid data.

Reset Sequence After the Dynamic Reconfiguration Process

This section describes the recommended reset sequence for your SATA and SAS channel after a dynamic reconfiguration of the channel to support different data rates. Depending on the reconfiguration method of your SATA and SAS transmitter or receiver, apply the appropriate reset sequence from one of the following recommendations.

Figure 6. Reset Sequence After Dynamic Reconfiguration—Data Rate Division in TX Mode

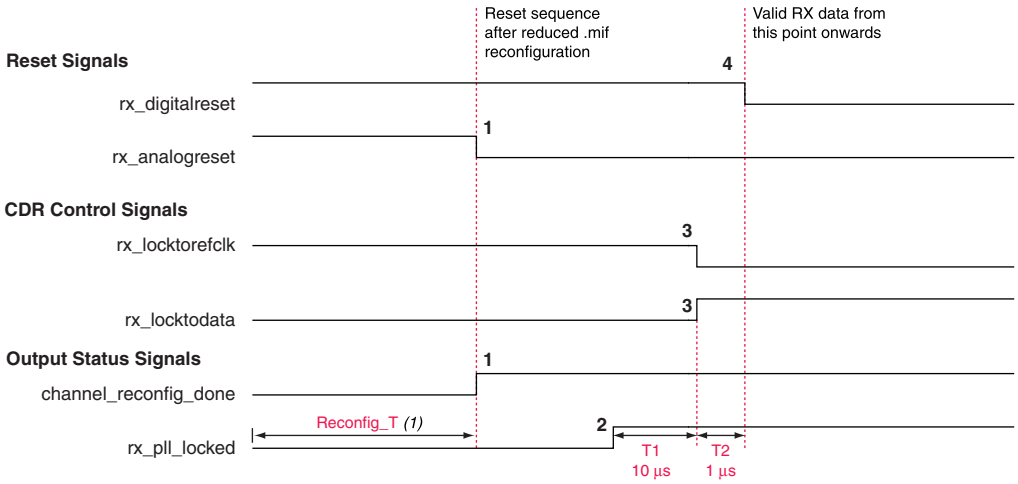


As shown in Figure 6, for the recommended reset sequence if you use the **Data Rate Division in TX** mode to reconfigure your transmitter channel, follow these reset steps:

1. After power up and properly establishing that the transmitter is operating as desired, write the desired 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.
2. Assert the `tx_digitalreset` signal.
3. As soon as `write_all` is asserted, the dynamic reconfiguration controller starts to execute its operation. This is 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. Lastly, `tx_digitalreset` can be de-asserted to continue with the transmitter operation (marker 4).

Figure 7 shows the timing diagram based on a receiver channel with the CDR configured in manual lock mode.

Figure 7. Reset Sequence After Reconfiguration—Reduced. mif Reconfiguration Mode



Note to Figure 7:

(1) The duration for the reconfiguration time (Reconfig_T) varies according to the device being used.

As shown in Figure 7, if you are reconfiguring the receiver channel using the **reduced .mif reconfiguration** mode, follow these steps:

1. After the `channel_reconfig_done` signal asserts high (indicating that the reduced **.mif** reconfiguration process is complete), deassert the `rx_analogreset` signal.
2. Monitor the `rx_pll_locked` signal. Once the signal goes high, wait for 10 µs, and then deassert the `rx_locktorefclk` signal and assert the `rx_locktodata` signal—to switch the CDR from the LTR to LTD mode.
3. Wait 1 µs and then deassert the `rx_digitalreset` signal. The receiver starts receiving valid data.

For the relock and reset scheme (if you use the reduced .mif reconfiguration mode), set the receiver CDR PLL to **Medium BW** (default) or **High BW**.

The entire reconfiguration, relock, and reset duration (including Reconfig_T, duration for the CDR to lock to data, and the reset sequence duration) must be within 54.6 µs. Ensure that the total time, in any reconfiguration mode, is within this restriction. As a general guide, the Reconfig_T should be less that 20 µs.

Timing Closure Involving Speed Negotiation

Timing closure is important and necessary if you are interfacing the transceiver to your design especially if you are performing speed negotiation in your SATA and SAS design.

As described in the previous section, you can use the ALTGX_RECONFIG controller to reconfigure your transceiver channel in SATA and SAS to switch dynamically between 1.5, 3, and 6 Gbps. The fitter in the Quartus II software tries to place and route the fabric to meet the data rate based on the constraints you specify.

If you intend to switch the transceiver between the three data rates, defining all clock rates and data paths on the transceiver core interface is crucial . You can use the TimeQuest timing analyzer to create the timing constraints that will help you to conform to the timing requirement for all your designs at the 1.5, 3, and 6 Gbps data rates. You must perform multiple clock constraints to ensure that you comply with the timing for all reconfigurable data rates in your design. This compliance allows for optimal logic placements for all data rates that are switched by the dynamic reconfiguration controller, without causing timing violations.

For more information about the design implementation and optimization, refer to the *Area and Timing Optimization* chapter in the *Quartus II Handbook Version 11.0*.

Document Revision History

Table 7 lists the revision history for this document.

Table 7. Document Revision History

Date	Version	Changes
June 2011	1.0	Initial release.