3.1 ALTGX_RECONFIG

Start the MegaWizard (Tools→MegaWizard Plug-In Manager)

- Page 1:
  - Select *Create a new custom megafunction variation.*
  - Click *Next.*
- Page 2a:
  - Select the *I/O→ALTGX_RECONFIG* component.
  - Select *VHDL* for the output file.
  - Name the component `altgx_rx_reconfig_x8`.
  - Click *Next.*
- Page 3 (see Figure 7)
  - Select 8 channels.
  - Uncheck *Analog controls.*
  - Check *Channel and TX PLL select/reconfig.*
  - Check *EyeQ control.*
  - Click *Next.*
- Page 5 (see Figure 8)
  - Check *Use 'reconfig_reset'.*
  - Click *Next.*
- Pages 6, 7, 8
  - Accept the defaults by clicking *Next*, and finally click *Finish.*

The Stratix IV Handbook does not describe the function of the `reconfig_reset` control, it just states that the control must be synchronous to `reconfig_clk` (Table 5-16, p935 [2]). The `reconfig_reset` can be used to delay the power-on offset cancellation until `reconfig_reset` is deasserted. Offset cancellation can also be delayed using the `offset_cancellation_reset` control. The offset cancellation input can only be added to the component using the `qmegawiz` command-line program per Knowledge Base entry rd09122011_555, whereas the `reconfig_reset` port can be added directly using the MegaWizard GUI. If the `reconfig_clk` is generated using a PLL, then offset cancellation *must* be delayed until the PLL output stabilizes per Knowledge Base entry rd12172009_309. If you look at the source code for the `altgx_rx_reconfig_x8` component, you will see that the `alt_cal` block reset input is driven by a combination of the reconfiguration and offset cancellation resets.

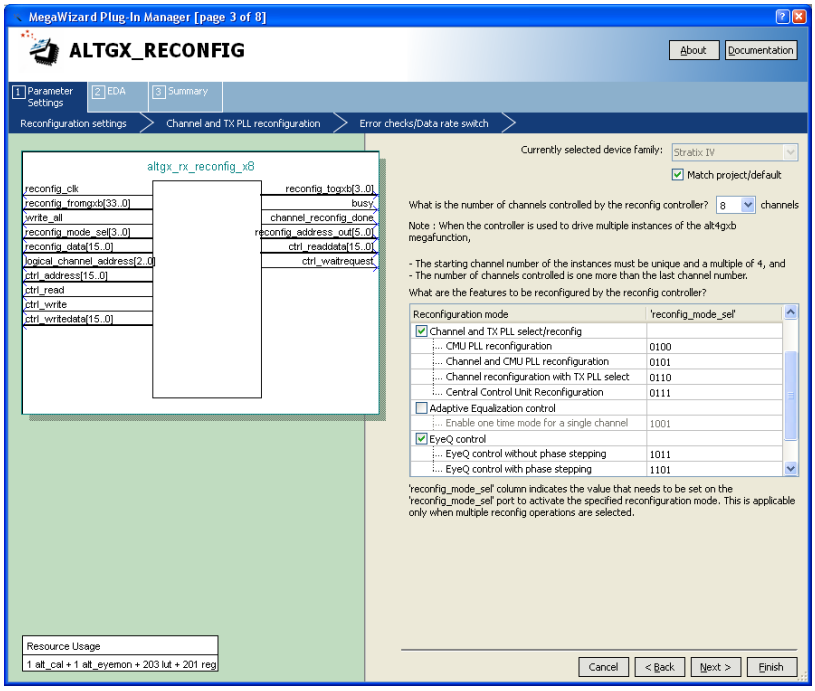

Figure 7: ALTGX_RECONFIG page 3.

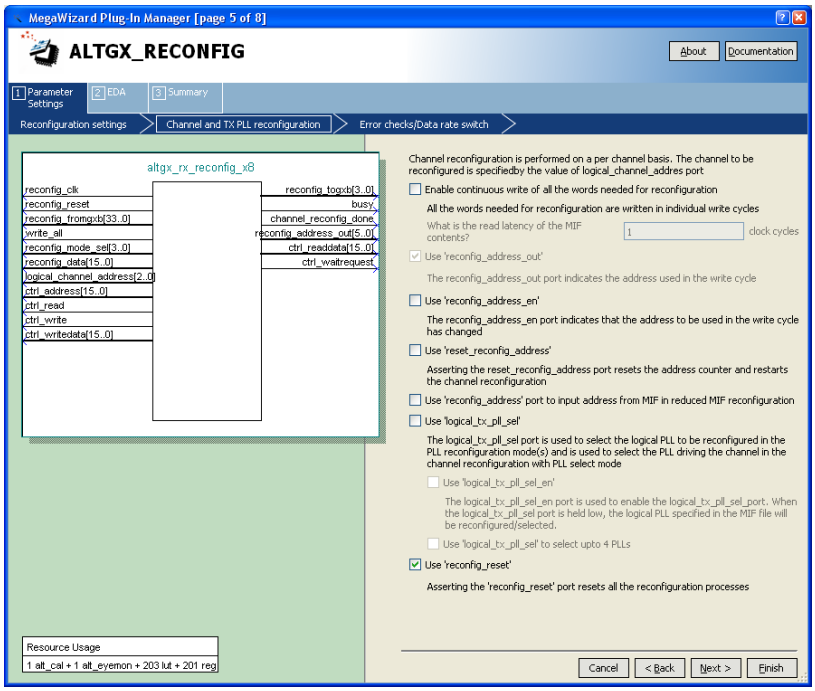

Figure 8: ALTGX_RECONFIG page 5.

3.2 Receiver-only ALTGX

Start the MegaWizard (Tools→MegaWizard Plug-In Manager)

- Page 1:
  - Select *Create a new custom megafunction variation.*
  - Click *Next.*
- Page 2a:
  - Select the *I/O→ALTGX* component.
  - Select *VHDL* for the output file.
  - Name the component `altgx_rx_x8_5000Mbps_32bits`.
  - Click *Next.*
- Page 3 (see Figure 9)
  - Select *Receiver only*
  - Select 8 channels.
  - Click *Double* for the deserializer block width
  - Select a channel width of 32-bits
  - Select a data rate of 5000Mbps
  - Select an input reference clock of 156.25MHz
  - Click *Next.*
- Page 4 (see Figure 10)
  - Check all of the *Optional Ports* (all that can be checked)
  - Click *Next.*
- Page 5 (see Figure 11)
  - Check *Create active high 'cal_blk_powerdown' port to powerdown the calibration block.*
  - Click *Next.*
- Pages 6, 7
  - Accept the defaults by clicking *Next.*
- Page 11 (see Figure 12)
  - Check *Analog controls (VOD, Pre-emphasis, Manual Equalization and EyeQ).*
  - Click *Next.*
- Page 21
  - Accept the defaults by clicking *Next.*
- Page 22 (see Figure 13)
  - Check *Use manual word alignment mode.*
  - Select a word alignment pattern length of 32-bits.

  - Enter an appropriate 32-bit pattern in binary format.
    * The Hittite bare die ADC board generates a repeating 16-bit XOR pattern of 00FFh (see Section 2.2). Configure the ALTGX 32-bit pattern as the binary value for 00FF_00FFh.
    * For a PRBS7 pattern, the 32-bit pattern (for the start of the PRBS) is the binary value for 8A18_207Fh.
  - Check *Create 'rx_patterndetect' output port for pattern detection*
  - Check *Create 'rx_invpolarity' to enable word aligner polarity inversion*
  - Check *Create 'rx_bitslipboundaryselectout' to indicate the number of bits slipped in the word aligner*
  - Click *Next*.

- Page 23 (see Figure 14)
  - Check *Enable byte ordering block*.
  - Select *The enabyteord signal from the PLD*.
  - Check *use a two word byte ordering pattern*.
  - Enter an appropriate 16-bit pattern in binary format.
    * For the Hittite bare die ADC board, the 16-bit pattern is the binary value for 00FFh.
    * For a PRBS7 pattern, the 16-bit pattern (for the start of the PRBS) is the binary value for 207Fh.
  - Enter 8-bits of zeros for the pad pattern.
  - Click *Next*.

- Pages 24 and 25
  - Accept the defaults by clicking *Next*, and finally click *Finish*.

TODO:

- The above shows the GX board 5000Mbps option.
- What does it take to get to 8500Mbps (GX device)?
- What about 10Gbps with the GT devices?
- How many of the ALTGX control bits do I actually use? Eliminate any that are not used.

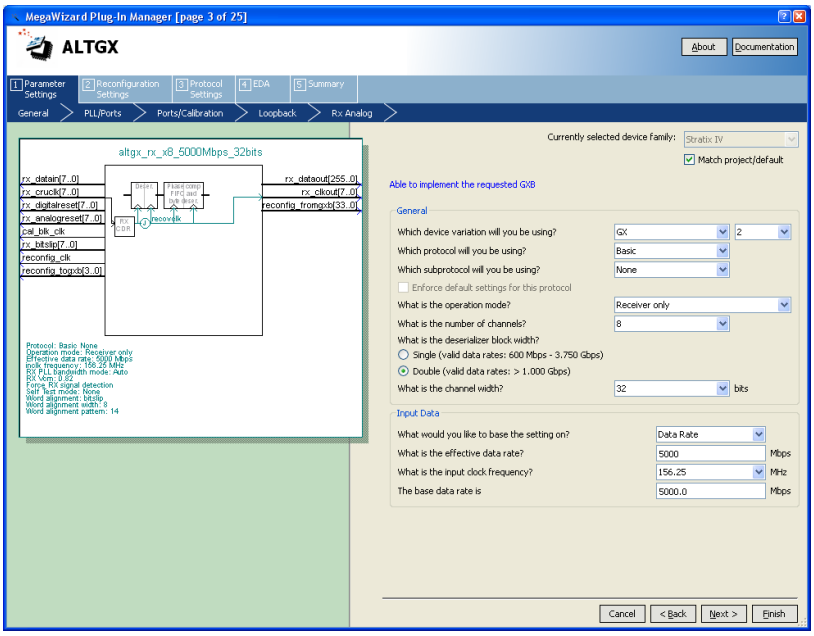

Figure 9: Receiver-only ALTGX page 3.

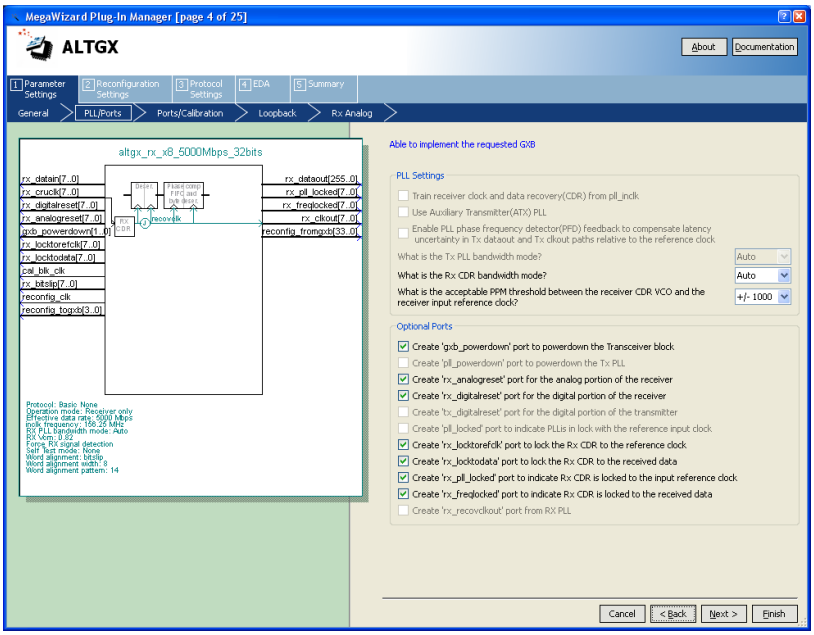

Figure 10: Receiver-only ALTGX page 4.

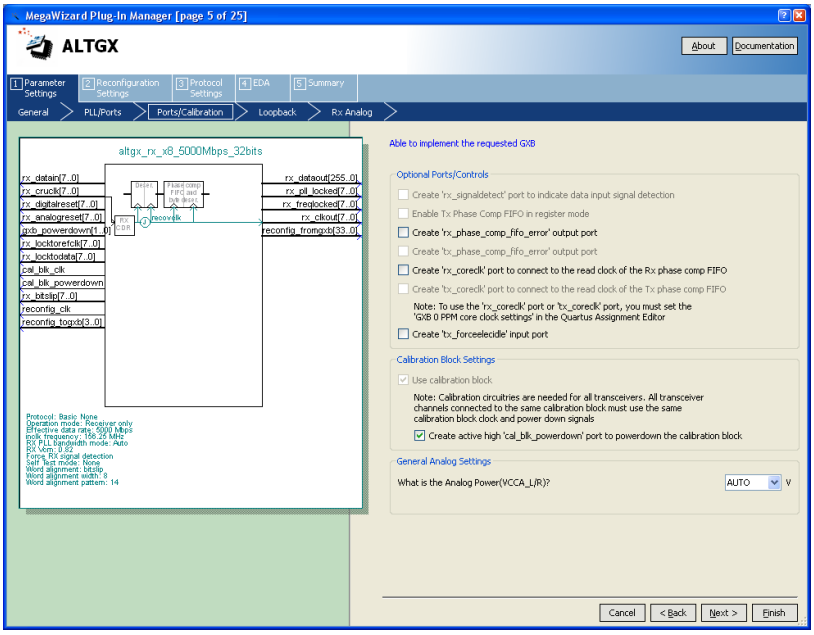

Figure 11: Receiver-only ALTGX page 5.

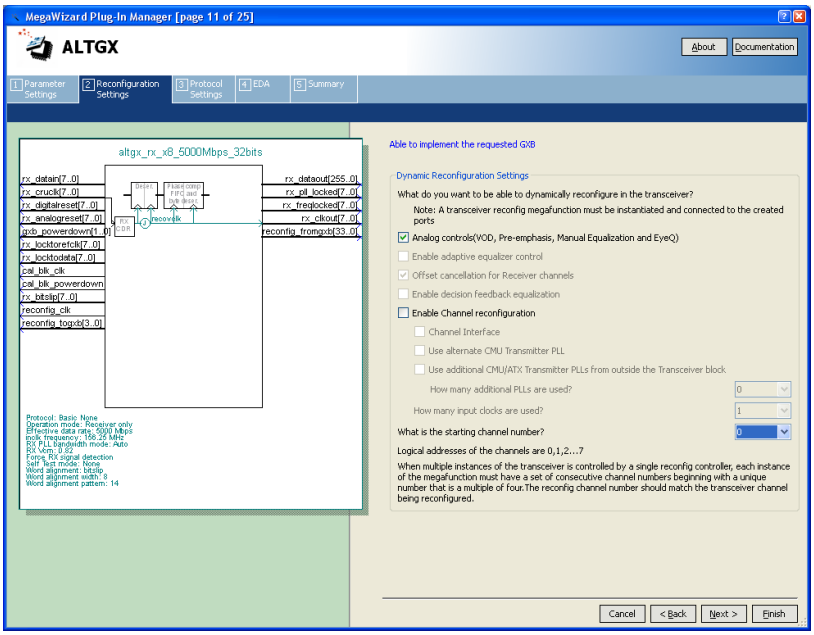

Figure 12: Receiver-only ALTGX page 11.

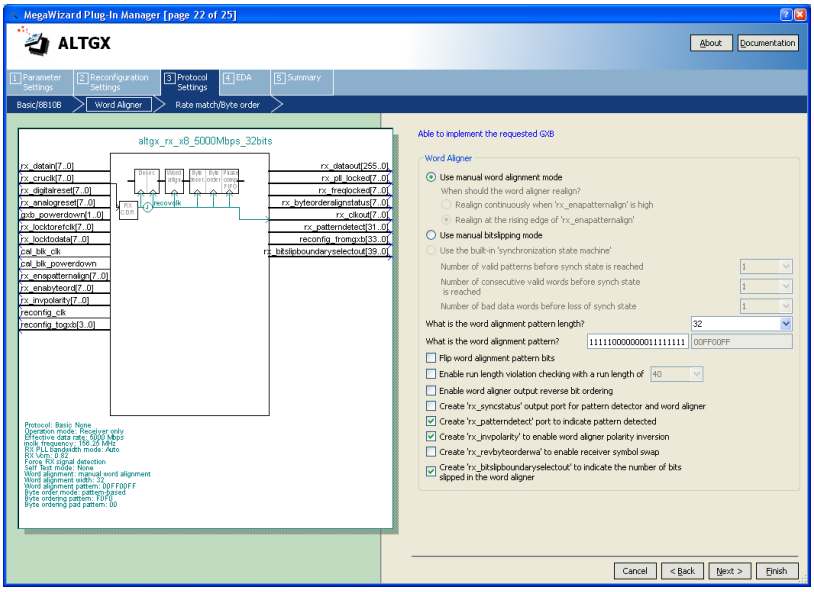

Figure 13: Receiver-only ALTGX page 22.

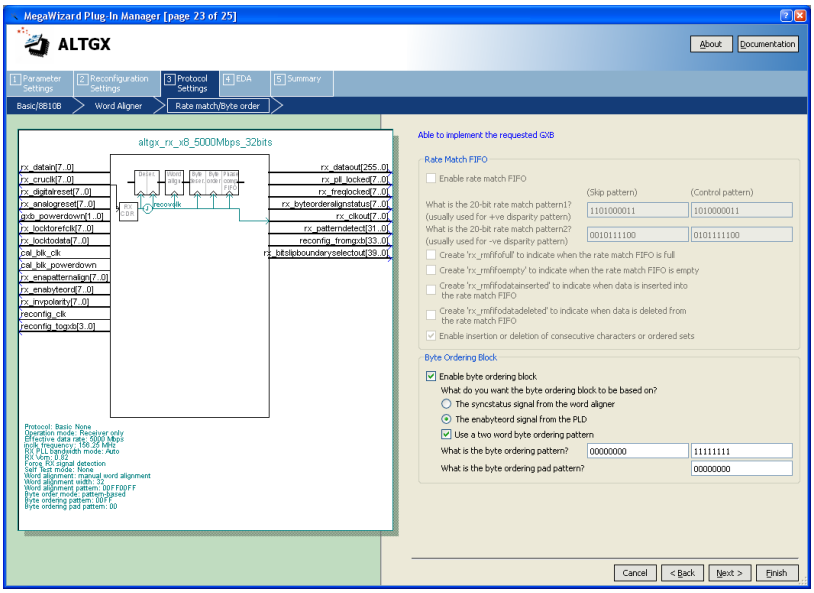

Figure 14: Receiver-only ALTGX page 23.

3.3 Receiver-only reset controller

The Altera transceiver reset requirements are described in the Stratix IV Handbook, Volume 2, Chapter 4 *Reset Control and Power Down in Stratix IV Devices* (p807 [2]). The `gxb_rx_reset_controller` component created for testing the Hittite ADC meets the transceiver reset requirements. This section shows SignalTap II logic analyzer traces captured from the receiver-only design (with the transceivers being driven by the ADC simulator).

General requirements

The general requirements for the receiver-only reset controller are;

- The ALTGX `gxb_powerdown` input must *not* be asserted when offset cancellation is running, i.e., when ALTGX_RECONFIG `busy` asserted (p914, p941 [2]). Once `busy` has deasserted, assert `gxb_powerdown` for at least 1μs (eg., see Figure 4-22, p844 [2]).
- The ALTGX_RECONFIG `busy` signal normally asserts at power-on. However, the discussion in Section 3.1 describes how the reconfiguration reset or offset cancellation reset ports can be enabled on the component. Asserting either of these reset signals delays offset cancellation until the reset signal is deasserted.

  This feature is useful when creating a component that holds a transceiver block in reset indefinitely, eg., statically asserting `reconfig_reset` means that `busy` will never assert, which allows `gxb_powerdown` to also be statically asserted (a `keep` synthesis constraint is required on the static `reconfig_reset` signal to stop Quartus II from eliminating what it thinks is unused logic).
- Table 4-4 in the Stratix IV Handbook shows the three different operating modes of the transceivers; manual lock-to-reference (LTR) mode, manual lock-to-data (LTD) mode, and automatic lock mode. The reset controller supports all three modes.

Lock-to-reference mode (LTR)

- Figures 15 to 17 show reset sequences for LTR mode.
- Figures 15 and 16 show the power-on assertion of the ALTGX_RECONFIG `busy` signal. The reset controller waits until `busy` deasserts before performing the reset sequence.
- Figure 17 shows the reset sequence performed again, after the power-on reset sequence has completed, i.e., the reset controller was enabled at power-on, disabled, and then re-enabled to produce the traces in the figure.
- The Stratix IV handbook does not show any receiver-only LTR mode timing diagrams. Figure 4-22 (p844 [2]) is the only figure that shows the `gxb_powerdown` signal. The timing diagrams for LTR mode are analogous to the start of the timing diagrams for LTD mode, without the transition to LTD mode (see the LTD traces).
- The reset sequence in Figures 16 and 17 proceeds as follows;
  - The user enables the controller for LTR mode (the `sys_rx_lock_to_refclk` and `sys_rx_lock_to_data` controls select the mode).
  - Wait for `reconfig_busy` to deassert.
  - Wait 1μs.
  - Assert `gxb_powerdown` for $t_{\texttt{gxb_powerdown}} = 1\mu$s (p1103, p1112 [2]).
  - Wait 1μs (the requirement for this delay is that it must be at least two clocks).
  - Deassert `gxb_rx_analog_reset`.
  - Wait for all receiver CDR PLLs to lock (as indicted by `gxb_rx_pll_locked`).
    The maximum lock time is $t_{\mathrm{LTR(max)}} = 75\mu$s (see the timing diagram Figure 1-2 on p1119, and the timing specifications on p1105 and p1114 [2]).
  - Wait $t_{\texttt{LTD_Manual}} = 4\mu$s
  - Deassert `gxb_rx_digital_reset`.
  - Assert `sys_rx_reset_done` to indicate the controller is finished.
- Figures 15 and 16 show the power-on assertion of the `reconfig_busy` signal. The `reconfig_count` was added to the design (and SignalTap II instance) to determine the length of the busy pulse. The busy pulse should be approximately 18,500 clocks per receiver channel plus 130,000, i.e., $8 \times 18500 + 130000 = 278000$ (43DF0h) or 5.56ms using a 50MHz clock (see the offset cancellation discussions on pp500-503, pp913-914, and the duration discussion on p939 [2]).

  Figure 16 shows a count of 483F4h (295924) which is a delay of 5.92ms. Different counts are obtained after each reconfiguration, but, the delay is always around 6ms.

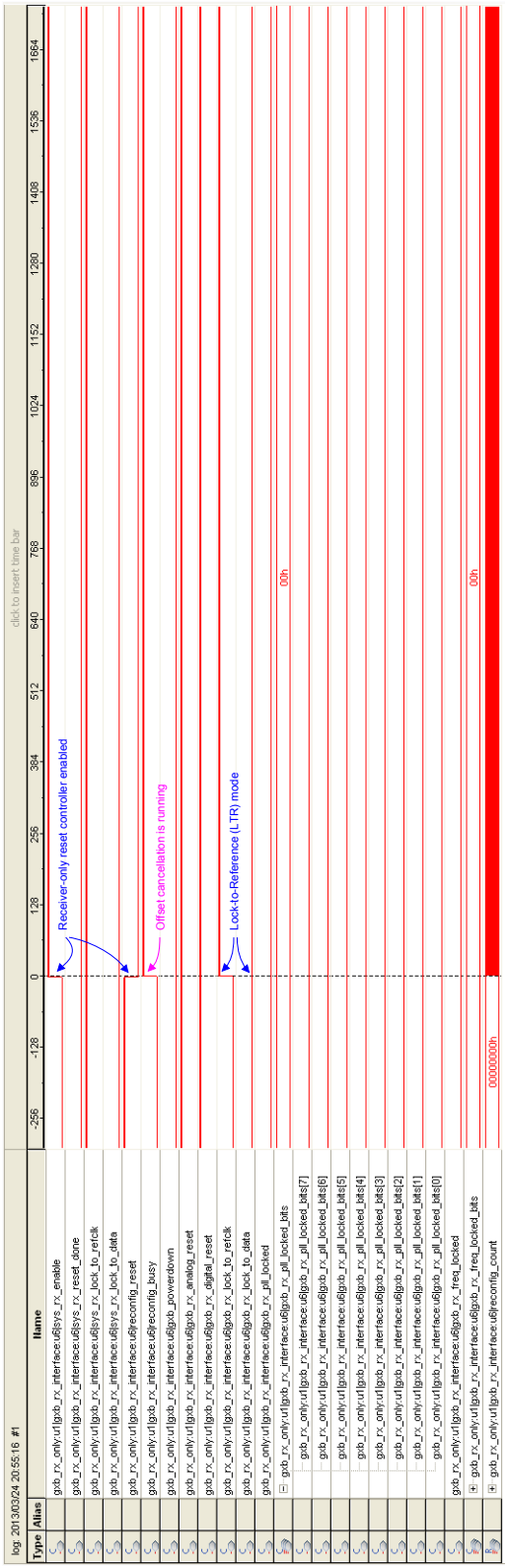

Figure 15: Altera Transceiver Receiver-only reset controller (LTR mode) power-on `reconfig_busy` assertion.

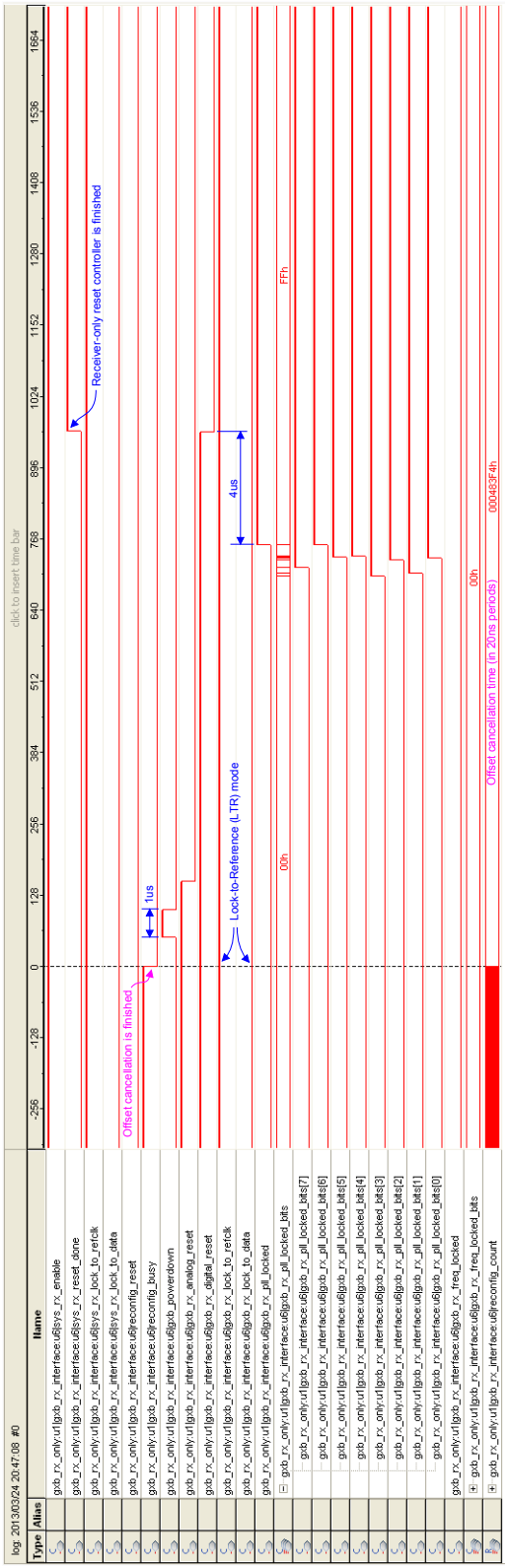

Figure 16: Altera Transceiver Receiver-only reset controller (LTR mode) power-on `reconfig_busy` deassertion.

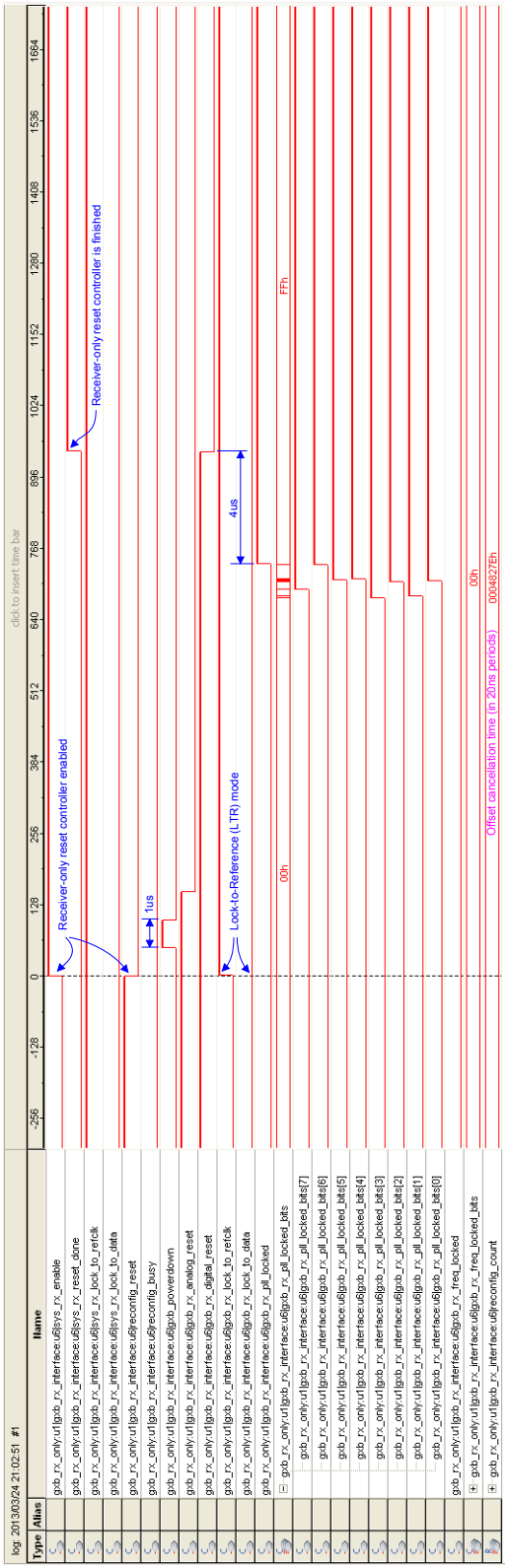

Figure 17: Altera Transceiver Receiver-only reset controller (LTR mode) re-enable (after power-on).

Lock-to-data mode (LTD)

- Figures 18 to 19 show reset sequences for LTD mode (for ALTGX_RECONFIG `busy` de-asserted).
- The Stratix IV Handbook [2] shows manual LTD mode timing diagrams for; four transmitter and receiver channels in Figure 4-5 (p816), eight transmitter and receiver channels in Figure 4-7 (p820), and a receiver-only channel in Figure 4-9 (p823). The SignalTap II traces in this section match those timing diagrams.
- The reset sequence in Figure 18 proceeds as follows;
  - The user enables the controller for LTD mode (the `sys_rx_lock_to_refclk` and `sys_rx_lock_to_data` controls select the mode).
  - Drive the transceiver `gxb_rx_lock_to_refclk` high and `gxb_rx_lock_to_data` low to start in LTR mode.
  - Wait for `reconfig_busy` to deassert.
  - Wait 1μs.
  - Assert `gxb_powerdown` for $t_{\texttt{gxb_powerdown}} = 1\mu$s (p1103, p1112 [2]).
  - Wait 1μs (the requirement for this delay is that it must be at least two clocks).
  - Deassert `gxb_rx_analog_reset`.
  - Wait for all receiver CDR PLLs to lock (as indicted by `gxb_rx_pll_locked`).

    The maximum lock time is $t_{\mathrm{LTR(max)}} = 75\mu$s (see the timing diagram Figure 1-2 on p1119, and the timing specifications on p1105 and p1114 [2]).
  - Wait $t_{\texttt{LTR_LTD_Manual}} = 15\mu$s. (p1105, p1114, p1119 [2]).
  - Drive the transceiver `gxb_rx_lock_to_refclk` low and `gxb_rx_lock_to_data` high to transition to LTD mode.
  - Wait $t_{\texttt{LTD_Manual}} = 4\mu$s
  - Deassert `gxb_rx_digital_reset`.
  - Assert `sys_rx_reset_done` to indicate the controller is finished.
- Figure 19 shows SignalTap II traces obtained when the receiver inputs were driven with static logic levels (similar waveforms occur when the HSMC-to-SMA connector is removed from the board).

  Note how the receiver reset sequence completes correctly, and that it is only after the reset controller is finished that the receiver PLL locked signals start to toggle.

  This trace shows that the system should monitor the receiver PLL lock signal while in LTD mode to determine whether receiver PLLs may have unlocked during LTD mode operation (this is the normal operating mode for data capture).

  Altera application note AN533: *Debugging Transceivers* indicates that PLL locked may toggle in lock-to-data mode (see p15 and 19 [1]), and that the frequency locked signal should deassert if the CDR loses lock-to-data. The LTD mode measurement in Figure 19, and the AUTO mode measurement in Figure 22, are inconsistent with these comments, in that the frequency locked signal remains asserted.

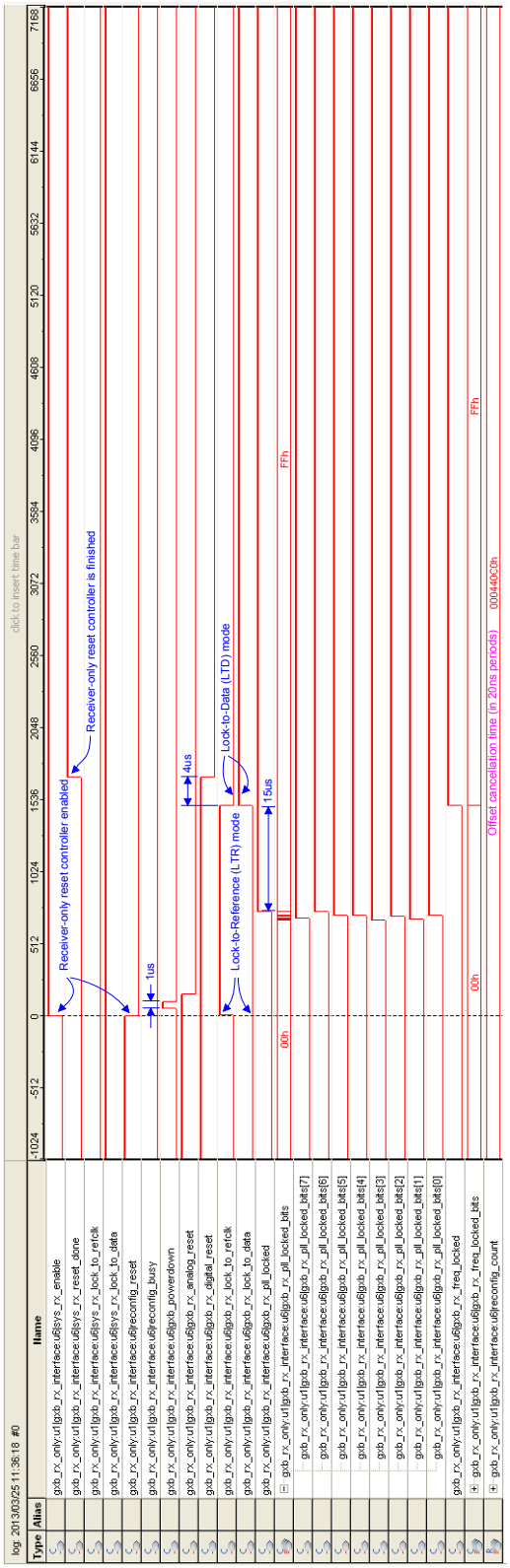

Figure 18: Altera Transceiver Receiver-only reset controller (LTD mode) re-enable after power-on.

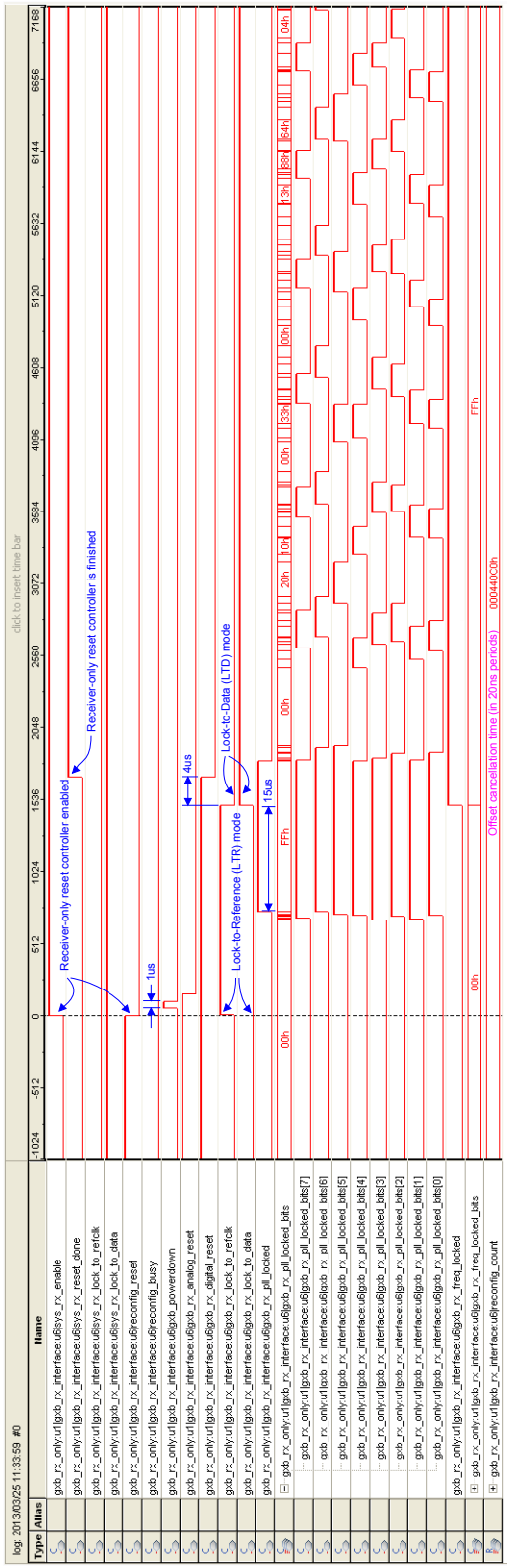

Figure 19: Altera Transceiver Receiver-only reset controller (LTD mode) re-enable after power-on, with no input data transitions.

Automatic lock mode (AUTO)

- Figures 20 and 21 show the reset sequence for AUTO mode (for ALTGX_RECONFIG `busy` deasserted).
- The Stratix IV Handbook [2] shows automatic mode timing diagrams for; four transmitter and receiver channels in Figure 4-4 (p814), eight transmitter and receiver channels in Figure 4-6 (p818), and a receiver-only channel in Figure 4-8 (p822). The SignalTap II traces in this section match those timing diagrams.
- The reset sequence in
- Figures 20 and 21 proceeds as follows;
  - The user enables the controller for AUTO mode (the `sys_rx_lock_to_refclk` and `sys_rx_lock_to_data` controls select the mode).
  - Drive the transceiver `gxb_rx_lock_to_refclk` and `gxb_rx_lock_to_data` both low to select AUTO mode.
  - Wait for `reconfig_busy` to deassert.
  - Wait 1μs.
  - Assert `gxb_powerdown` for $t_{\texttt{gxb_powerdown}} = 1\mu$s (p1103, p1112 [2]).
  - Wait 1μs (the requirement for this delay is that it must be at least two clocks).
  - Deassert `gxb_rx_analog_reset`.
  - Wait for all receiver CDR PLLs to lock (as indicted by `gxb_rx_pll_locked`).
    The maximum lock time is $t_{\text{LTR(max)}} = 75\mu$s (see the timing diagram Figure 1-2 on p1119, and the timing specifications on p1105 and p1114 [2]).
  - Wait for all receiver CDR PLLs to frequency lock (as indicted by `gxb_rx_freq_locked`).
    The `freq_locked_count` was added to the design to measure the time between the assertion of `gxb_rx_pll_locked` and `gxb_rx_freq_locked`. Multiple reset trace captures showed the count was consistently 50DDh or 50DEh, i.e., about 20700 clocks, or 414μs. The Stratix IV Handbook has no details about this timing parameter.
    Altera application note AN533: *Debugging Transceivers* (p10 [1]) discusses what occurs between PLL lock and frequency lock; the CRU compares the recovered clock with the reference clock, and when the parts per million (PMM) detector detects that the difference is within that specified for the ALTGX instance, it switches to the recovered clock and asserts the frequency locked indicator. At this point, the recovered clock is frequency locked, but not phase-locked. The 4μs delay after frequency lock allows for phase-lock.
    Figure 8 in AN533 shows a SignalTap II trace from a Stratix IV GX transceiver for automatic lock mode (the SignalTap II instance uses a 20MHz clock, with 50ns period). The PLL and frequency locked signals assert about 8500 clocks apart or 425μs. This is consistent with the measurement in Figure 21.
  - Wait $t_{\texttt{LTD_Auto}} = 4\mu$s
  - Deassert `gxb_rx_digital_reset`.
  - Assert `sys_rx_reset_done` to indicate the controller is finished.
- Figure 22 shows the end of the AUTO reset sequence with the HSMC-to-SMA adapter removed, i.e., nothing attached to the receiver inputs. Note how the PLL locked signals toggle, however, the frequency locked signal remains asserted. This behavior is inconsistent with AN553's description of the operation of the frequency locked signal, and example in Figure 11 on p15 [1].

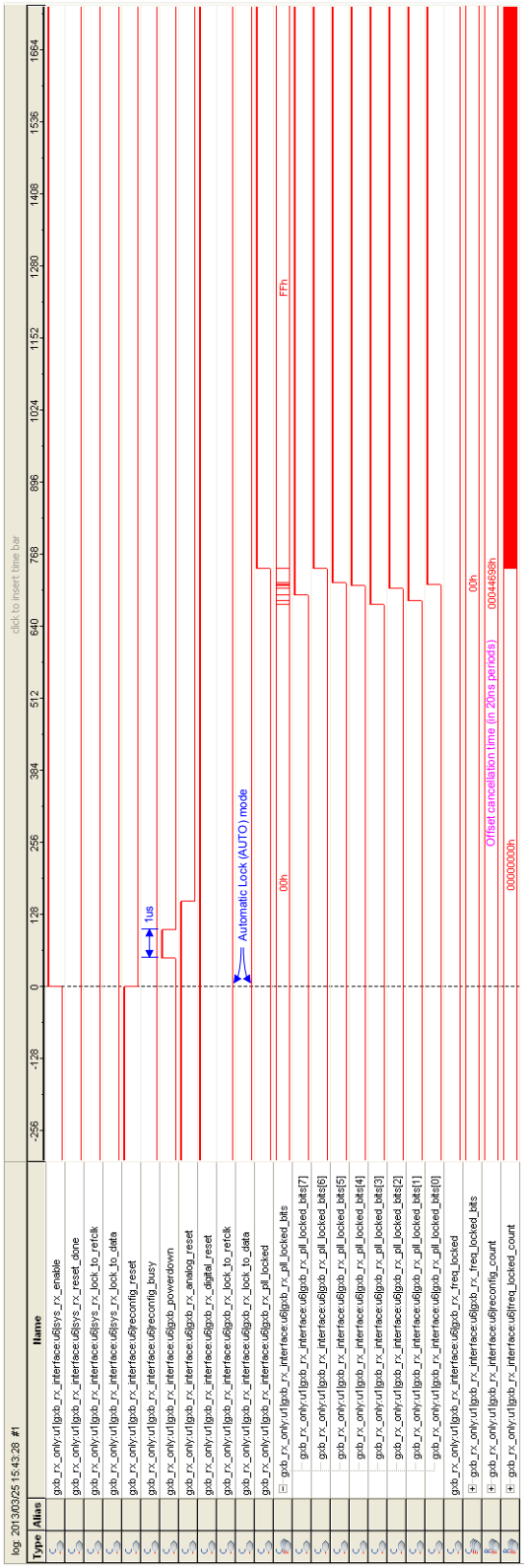

Figure 20: Altera Transceiver Receiver-only reset controller (AUTO mode) re-enable after power-on (start of reset sequence).

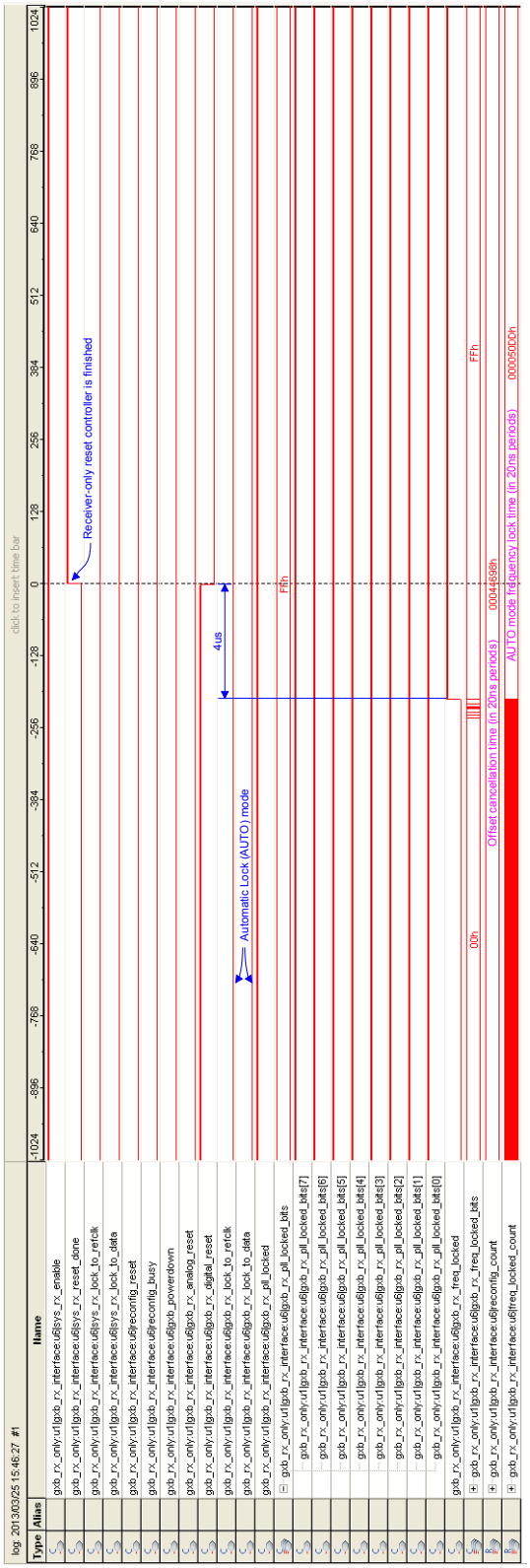

Figure 21: Altera Transceiver Receiver-only reset controller (AUTO mode) re-enable after power-on (end of reset sequence).

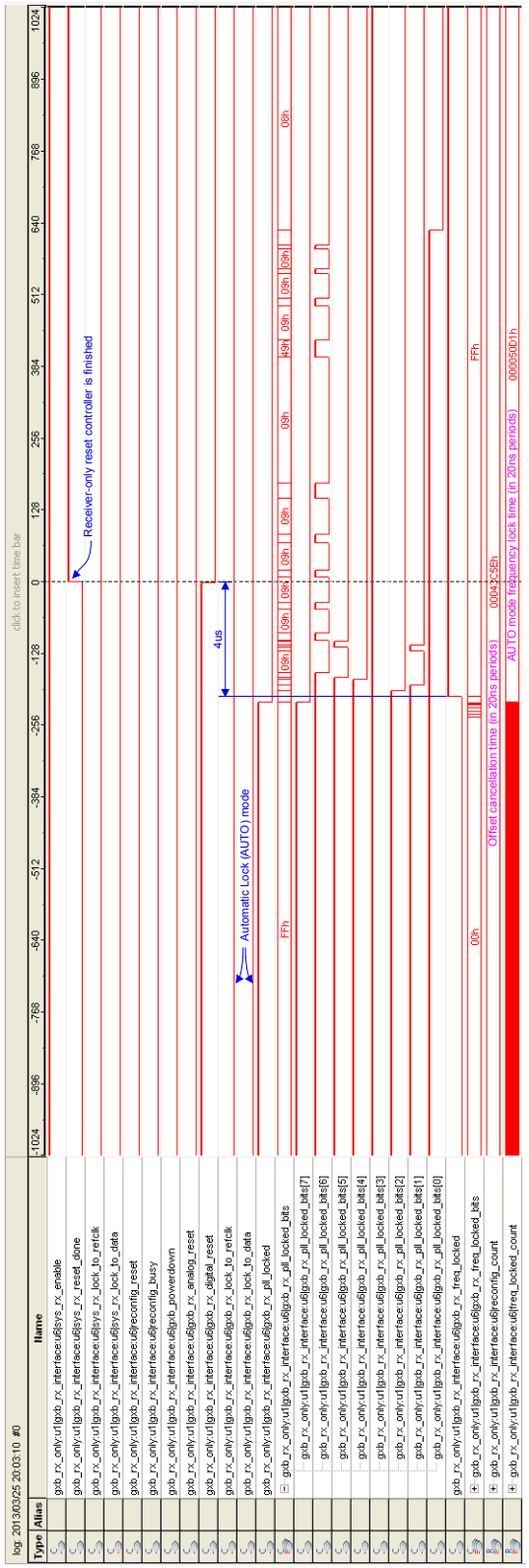

Figure 22: Altera Transceiver Receiver-only reset controller (AUTO mode) re-enable after power-on (end of reset sequence), with no input data transitions.

3.4 Transmitter-only reset controller

References

[1] Altera Corporation. AN553: Debugging Transceivers (version 1.1), December 2009. (an553.pdf).

[2] Altera Corporation. Stratix IV Handbook (version 4.6), September 2012. (stratix4_handbook.pdf).