SYNOPSYS®

DesignWare® Cores
Multi-Protocol 32G PHY ATE Testbench

Application Note

Doc. Version: 1.30a
August 23, 2021

Copyright Notice and Proprietary Information

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

Synopsys and certain Synopsys product names are trademarks of Synopsys, as set forth at https://www.synopsys.com/company/legal/trademarks-brands.html

All other product or company names may be trademarks of their respective owners.

Free and Open-Source Software Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

Synopsys, Inc.
www.synopsys.com

Contents

Revision History

The following table lists the revision history of the PHY for each release. For a detailed description of PHY component updates, refer to the PHY release notes.

Note In some instances, documentation-only updates occur. The DesignWare IP product on https://www.synopsys.com/designware-ip.html has the latest information.

Date	Version	Description
August 23, 2021	1.30a	Removed: ■ A footnote “Tested only when baud rate is >= 8 Gbps” throughout “ATE Testing Details”
July 29, 2021	1.20a	Updated: ■ “TX DC Levels Test”, - External loopback required: No - Wafer Sort: Yes ■ “Common Mode Test” - External loopback required: No - Wafer Sort: Yes
September 2020	1.10	Updated: ■ “Power-up Calibrations” ■ Table 1-1, “Calibration Time” ■ “ATE Testbench and Simulation” ■ Table A-1, “ATE Override Control Pin List for SERDES_32G_PHY and PIPE”
April 2020	1.0	The changes since the previous version include the following: Added: ■ Note in “ATE Test Suite” Updated: ■ Table A-1, “ATE Override Control Pin List for SERDES_32G_PHY and PIPE”

Date	Version	Description
January 2020	0.40	The changes since the previous version include the following: Updated: ■ Links to https://solvnetplus.synopsys.com throughout the book ■ Links to web resources in “Preface” ■ “Customer Support”
October 2019	0.30	Removed: ■ Firmware disable mode throughout the Application Note Updated: ■ Table A-1(“ATE Override Control Pin List for SERDES_32G_PHY and PIPE”)
May 2019	0.20	Added: ■ “ATE Testbench and Simulation” section ■ Note about Register Test in Table 2-3 (“ATE Tests”) ■ Note after Table A-1 (“ATE Override Control Pin List for SERDES_32G_PHY and PIPE”) Updated: ■ Table A-1 (“ATE Override Control Pin List for SERDES_32G_PHY and PIPE”)
January 2019	0.10	Initial version

Preface

This document describes the ATE Verilog Testbench that enables you to generate test and characterization vectors and perform ATE testing on SERDES_32G_PHY macros. In addition, this document provides the procedure for each test in the ATE test suite.

Application Note Organization

This document is organized as follows:

- Chapter 1, "ATE Verilog Testbench Overview", provides an introduction to the ATE Verilog Testbench and its features.
- Chapter 2, "ATE Testing Details", describes the test, load board, and test mode pin requirements as well as a description of the ATE vectors, test suite, test sequence, and coverage.
- Appendix A, "ATE Override Control Pin List", lists the initial conditions for the pins.

Web Resources

- DesignWare IP product information: https://www.synopsys.com/designware-ip.html
- Your custom DesignWare IP page: https://www.synopsys.com/dw/mydesignware.php
- Documentation through SolvNetPlus: https://solvnetplus.synopsys.com (Synopsys password required)
- Synopsys Common Licensing (SCL): https://www.synopsys.com/keys

STARs on the Web (SotW)

You must review all STARs on the Web (SotWs) associated with your product. SotWs are considered a part of the Synopsys documentation suite, and show critical information related to your product. To review product SotWs, refer to the DesignWare IP product information:

https://www.synopsys.com/designware-ip.html

Synopsys Statement on Inclusivity and Diversity

Synopsys is committed to creating an inclusive environment where every employee, customer, and partner feels welcomed. We are reviewing and removing exclusionary language from our products and supporting customer-facing collateral. Our effort also includes internal initiatives to remove biased language from our engineering and working environment, including terms that are embedded in our software and IPs. At the same time, we are working to ensure that our web content and software applications are usable to people of varying abilities. You may still find examples of non-inclusive language in our software or documentation

as our IPs implement industry-standard specifications that are currently under review to remove exclusionary language.

Customer Support

To obtain support for your product, contact Support Center using one of the following methods:

- *For the fastest response*, enter a case through SolvNetPlus:

  a. https://solvnetplus.synopsys.com

  Note SolvNetPlus does not support Internet Explorer. Use a supported browser such as Microsoft Edge, Google Chrome, Mozilla Firefox, or Apple Safari.

  b. Click the **Cases** menu and then click **Create a New Case** (below the list of cases).

  c. Complete the mandatory fields that are marked with an asterisk and click **Save**.

  Make sure to include the following:

  - **Product L1:** DesignWare Cores
  - **Product L2:** <name of L2>

  d. After creating the case, attach any debug files you created.

  For more information about general usage information, refer to the following article in SolvNetPlus:

  https://solvnetplus.synopsys.com/s/article/SolvNetPlus-Usage-Help-Resources

- Or, send an e-mail message to support_center@synopsys.com (your email will be queued and then, on a first-come, first-served basis, manually routed to the correct support engineer):
  - Include the Product L1 and Product L2 names, and Version number in your e-mail so it can be routed correctly.
  - For simulation issues, include the timestamp of any signals or locations in waveforms that are not understood
  - Attach any debug files you created.
- Or, telephone your local support center:
  - North America:
    Call 1-800-245-8005 from 7 AM to 5:30 PM Pacific time, Monday through Friday.
  - All other countries:

  https://www.synopsys.com/support/global-support-centers.html

1 ATE Verilog Testbench Overview

This chapter includes the following sections:

1.1 Introduction

Developing a production test for an ASIC is becoming more complicated. ASICs contain more functions than previously, and many of the functions are purchased as IP from outside sources. Test engineers must understand these IP functions to test them effectively. To compound the issue, test engineers usually have little or no direct contact with the people who developed these functions. Digital scan tools have been developed to solve part of this problem. These scan tools can develop test vectors to test the digital portion of ASICs without users knowing details about the function of the circuits tested. These scan tools can compute how effectively the vectors test the digital portion of the IP. Often, these scan patterns are delivered with the IP to further simplify the matter.

The challenge is testing analog functions that digital scan does not test effectively. There are no industry tools available to assist with this task. Test engineers have to develop and implement a test plan that will effectively and efficiently test the functions. Test engineers must use their knowledge, experience, and information they can gather to develop the test plan. This task can be daunting if many different pieces of IP are on the ASIC with which the test engineers are not familiar. After the test plan is developed, the test must be implemented for the specific tester using the appropriate tester analog resources available. The whole analog test development process can be difficult and labor intensive.

For analog testing, vectors are scanned in and scanned out of a JTAG port, similar to a digital scan. No complicated test code or analog test resources are required. Analog testing is a simple pass/fail operation.

1.2 Methodology

Typically, each test is configured through the JTAG port by writing internal IP registers. Resulting values in IP registers are compared and a pass/fail value is returned through the JTAG port. In the Synopsys solution, no vector capture is required. Every test returns a simple pass/fail result.

1.3 Built-In Features

Test features were designed using the PHY's existing circuitry, enabling the design to minimize the impact to die size. This section describes the test features that are included in the IP.

1.3.1 Pattern Generator

The SERDES_32G_PHY includes a pattern generator that enables several different patterns to be sent out of the transmitter during testing.

1.3.2 Pattern Matcher

The SERDES_32G_PHY provides a pattern matcher that is used to verify the transmit-to-receive path of the IP.

1.3.3 JTAG Accessibility

JTAG access is used to read and write registers in the IP, as detailed in the DesignWare Cores SERDES_32G_PHY databook.

The JTAG interface is enabled when input cr_para_sel is de-asserted to 0.

1.3.4 Control Register (CR) Parallel Accessibility

The Control Register (CR) Parallel register interface can be used to read and write registers in the IP, as detailed in the DesignWare Cores SERDES_32G_PHY databook. The CR interface is enabled when input cr_para_sel is asserted to 1. Refer to phy/testbench/ATE_TESTBENCH.README ACCESS_TYPE for control variable values to select JTAG or CR access within the ATE testbench.

1.3.5 Analog Test Bus (ATB) and Digital Test Bus (DTB) Signals

The analog test bus (ATB) provides a way to access internal voltages for debug purposes. When integrating the PHY, the following ATB pins should be left unconnected:

- atb_s_p
- atb_s_m
- atb_f_p

The digital test bus (DTB) provides a way to access internal digital signals for debug purposes. If the DTB is unneeded, the DTB pins (dtb_out[1:0]) may be left unconnected when integrating the PHY.

1.3.6 DAC/ADC

An on-board 10-bit DAC/ADC is included in the PHY IP. The DAC can be applied to the test bus to force internal nodes and it also enables the measurement of internal nodes.

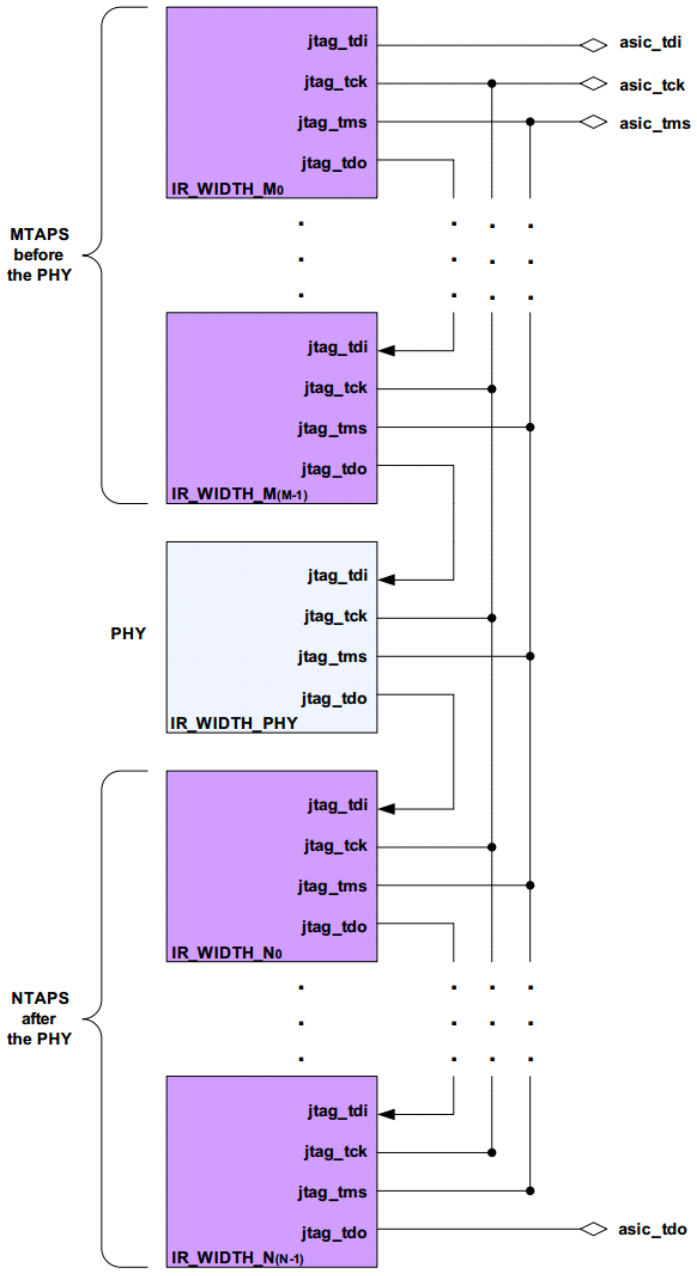
To minimize the effect on area, this circuit reuses features of the transmitter and receiver termination calibration circuitry.

1.4 JTAG Configurability

The ATE Verilog Testbench can be configured to support multi-TAP configurations and different IR widths in each TAP controller as shown in Figure 1-1.

To configure the ATE Verilog Testbench, the JTAG chain information must be entered in the tb_tap_config.v and tb_ir_width_override.v files (located in `macro/Latest/testbench/tb_ate_jtag_driver.v`).

Figure 1-1 JTAG Configuration



1.5 Example Configuration

This section provides an example configuration and the steps needed to configure the ATE Verilog Testbench.

In the example, the following configuration is required:

- MTAPS = 1
- IR_WIDTH_M0 = 8
- NTAPS = 0
- IR_WIDTH_N0 = 0

The required steps are as follows:

1. Modify the default branch's content of the tb_tap_config.v file as follows:

```
`define MTAPS 1 //Number of tap controllers before the PHY
`define IR_WIDTH_M0 8 //Width of IR for tap controller 0 before the PHY
`define NTAPS 0 //Number of tap controllers after the PHY
`define IR_WIDTH_N0 0 //Width of IR for tap controller 0 after the PHY
`define NONZEROTC_M //Must be defined if MTAPS >= 1
`define MTAPS_IRW_SUM 8 //Sum of width of IRs of all the tap controllers before the PHY
//`define NONZEROTC_N //Must be defined if NTAPS >= 1 [Commented out since NTAPS = 0
`define NTAPS_IRW_SUM 0 //Sum of width of IRs of all the ta]p controllers after the PHY
```

2. Modify the default branch's content of the tb_ir_width_override.v file as follows:

```
defparam mtaps[0].jtag_m.JTAG_IR_WIDTH = `IR_WIDTH_M0;
```

3. Run simulation with the ATE Verilog Testbench.

The TAP controllers before and after the PHY will be placed in BYPASS mode.

1.6 Vector Generation

To generate a vector file from the ATE Verilog Testbench, enable the VEC_TRACE define and disable the <PREFIX>_SHORT_RESET define at run time.

<PREFIX> used for this product can be found in the <install_directory>/<rel_version>/phy/testbench/TESTBENCH_PHY_ATE.README file.

For this, the following snippet of code can be used:

```
`ifdef VEC_TRACE
`undef <PREFIX>_SHORT_RESET
`endif
```

The following example runs all regression tests for PHY interface:

```
run_ate_regression.sh -proto phy -sim_type gtech -sim_tool vcs -grid 0
```

The following example runs all regression tests for PIPE interface:

```
run_ate_regression_upcs.sh -proto upcs -sim_type gtech -sim_tool vcs -grid 0
```

The vector file is named ate.vec, and it contains bitstreams in the tms_tdi_tdo format. Each line in the vector file corresponds to a positive edge of the JTAG clock (tck). The ATE Verilog Testbench internally masks and unmasks the tdo output.

As part of the simulation run, for either individual tests or the full regressions, tests are run with the raw-PCS firmware FSM enabled.

When the raw-PCS firmware FSM is enabled and the power-up sequence is run, the following additional tasks are performed along with the tasks listed above, thereby improving the coverage at the cost of increased tester time:

- RX offset cancellation
- RX slicer calibration
- RX Equalization and Adaptation (as applicable)

1.6.1 Power-up Calibrations

When the raw-PCS firmware FSM is enabled and the power-up sequence is run, RX start-up calibration can take up to T_{normal} ms (the calibration time for normal FW where <PREFIX>_FW_NORMAL_SIM is defined), which can mean very long simulation times.

To reduce simulation run times, define the <PREFIX>_SHORT_RESET flag for vector generation. This action not only reduces the calibration time during vector generation simulations but also is still fully compatible with normal mode firmware. In fact, when defining the <PREFIX>_SHORT_RESET, the tb_ate_tasks.v will automatically insert dummy cycles in ate.vec that match the correct ate.vec. We recommend using <PREFIX>_SHORT_RESET to generate ate.vec

Table 1-1 Calibration Time

Number of Lanes	Reference Clock Frequency (MHz)	Calibration Time (ms)
		T_{normal} (<PREFIX>_FW_FAST_SIM not defined)
x1/x2/x4	<= 40	19
	> 40	15

1.6.2 Startup Adaptation

When the raw-PCS firmware FSM is enabled, RX Equalization and Adaptation is automatically performed only when external loopback configuration is used (<PREFIX>_WAFER_SORT is undefined) and the baud rate is 8 Gbps or more. To override this constraint, set the following in the ate_param_settings.v file:

```
ate_adapt_reqd=1'b0;
```

1.6.3 External SRAM Loading

By default, the firmware stored in the ROM (either external or internal) is executed during the ATE sequence. To use the firmware stored in the ROM, you can tie the sram_ext_ld_done to 1'b1 to denote that there will be no external firmware loading when performing ATE at the PHY level.

When performing ATE at the PIPE level however, you must tie the phyP_sram_ext_ld_done signal to 1'b0. This action is required to allow the ATE sequence to perform necessary configuration updates.

For external loading of firmware, tie the sram_ext_ld_done (phyP_sram_ext_ld_done signal at the PIPE level) to 1'b0 when the phy_reset signal is de-asserted. This action puts the PHY/PIPE into an external firmware loading mode. At this point, there are two options to support externally loading firmware. The options are outlined as follows:

- Sideload Updated Firmware
- Load Updated Firmware Through the PHY

1.6.3.1 Sideload Updated Firmware

Use this option to externally 'sideload' the SRAM through either a different SRAM port or through muxes on the SRAM write interface

ATE vectors are generated as they normally would, but they are then post-processed to add an appropriate wait time during the external loading mode while the SRAM is sideloaded. The following statement can be found in the log file:

```
Waiting for 2 microseconds for expected events to take place
Insert the vector pause for FW updates if necessary in ate.vec at line
number 3591
Waiting for 2 microseconds for expected events to take place
```

1.6.3.2 Load Updated Firmware Through the PHY

This option applies to the case where the updated firmware is loaded through the PHY CR PARA or JTAG interface.

In this case, you should manually modify the tb_ate_tasks.v file to insert register writes to load the firmware through the appropriate interface. Loading the firmware can also be done using the internal asr() function used throughout the tb_ate_tasks.v file for performing register writes. The modifications should be done at the 'Update firmware if necessary' comment within the tb_ate_tasks.v file.

1.7 Verilog Testbench Configurability

This section provides information about configuring the ATE Verilog Testbench.

1.7.1 JTAG Clock Frequency

The default JTAG clock frequency is 16.666 MHz. As shown in the following snippet of code, this frequency is specified in the Verilog file, macro/Latest/testbench/tb_ate_jtag_driver.v, using the tck_phase parameter.

```
//------------------------------------------------------------------------
// Clock Driver
//------------------------------------------------------------------------
parameter  JTAG_TCK_PHASE = 30;
real tck_phase
initial begin
  tck_pin = 0;
  tck_phase = JTAG_TCK_PHASE;
  forever begin
    if(installed) begin
      #(tck_phase) tck_pin = ~tck_pin;
    end
    else begin
      // need something to not have 0 delay loop
      #(tck_phase);
    end
  end
end
```

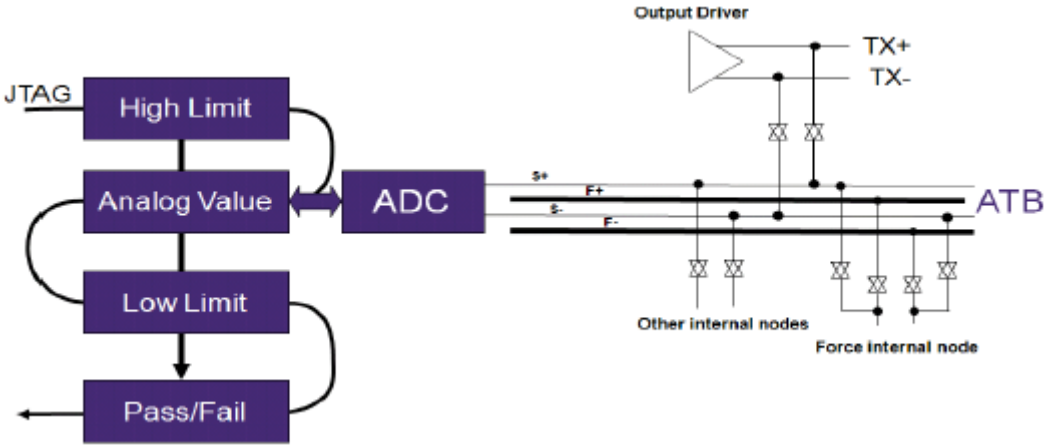
Jtag_clock frequency = 1 / (2 * tck_phase). For instance, to set the JTAG clock frequency to 10 MHz, set tck_phase to 50 (assuming time unit is 1 ns).

1.8 Example Test

The TX DC Levels test is a good example to illustrate features inside the ATE Verilog Testbench IP in use.

In this test, the pattern generator is set up to send an “all 1's” pattern with maximum and minimum voltage settings, the ADC is attached to the ATB, and status is read through the SUP_DIG_RTUNE_STAT register. The pattern generator is then set up to send an “all 0's” pattern with maximum and minimum voltage settings, the ADC is attached to the ATB, and status is read through SUP_DIG_RTUNE_STAT register. Figure 1-2 shows an example of the TX DC Levels test setup.

Figure 1-2 Example of TX DC Levels Test



1.9 ATE Testbench and Simulation

The ATE supports both the UPCS and PHY level testbench and simulation.

Testbench and Readme files can be found in ./Latest/phy/testbench and ./Latest/upcs/testbench:

- tb_ate_top.v - ATE top level file
- tb_ate_tasks.v - ATE testbench tasks
- ATE_TESTBENCH.README - ATE testbench readme file

Simulation script can be found in ./Latest/phy/sim and ./Latest/upcs/sim:

- run_ate_regression.sh or run_ate_regression_upcs.sh - ATE regression script file
- run_ate_test.sh or run_ate_test_upcs.sh - ATE run script

Refer to the ATE_TESTBENCH.README for running the ATE testbench.

Table 1-2 ATE Simulation Mode (fw_sim_mode Option in the Regression Script)

Simulation Mode	Macro Defined in the Regression Script	Description
short	{PREFIX}_SHORT_RESET	Short simulation mode. The calibration sequence is skipped for short simulation and the simulation takes short time.
normal	{PREFIX}_ FW_NORMAL_SIM	Normal firmware mode. The full sequence of the firmware.

2 ATE Testing Details

This chapter includes the following sections:

2.1 Tester Requirements

The SERDES_32G_PHY test methodology uses minimal tester resources. The tester's hardware requirements include two power supplies, four digital pins, and a differential reference clock. In most applications, two digital tester pins can be used as the reference clock source. The reference clock must be present before the PHY is released from reset.

Table 2-1 lists the minimum tester requirements.

Table 2-1 Minimum Tester Requirements

Function	Details
Power Supply	For power requirements, refer to the DesignWare Cores SERDES_32G_PHY databook.
Digital force (JTAG TDI, TMS, and TCK)	Three pins typically at 10 MHz
Digital compare (TDO)	One pin typically at 10 MHz
Reference clock	100 MHz
Initial pin settings and tie-offs	See “ATE Override Control Pin List” on page 51

2.2 Load Board Requirements

Load board requirements are minimal. The PHY requires an external 200 Ω resistor on the resref pin. To take advantage of the built-in test capabilities, an external AC-coupled loopback between the SERDES_32G_PHY transmit and receive pins is required. If continuity/leakage measurements are required on the serial interface, place an optional relay on the signal path. To minimize any discontinuities in the signal path, take care when placing the relay. If only a continuity check is required, you can place a high-value resistor on the loopback path.

2.3 Test Mode Pin Requirements

Figure 2-1 shows the external board requirements.

Figure 2-1 External Board Requirements

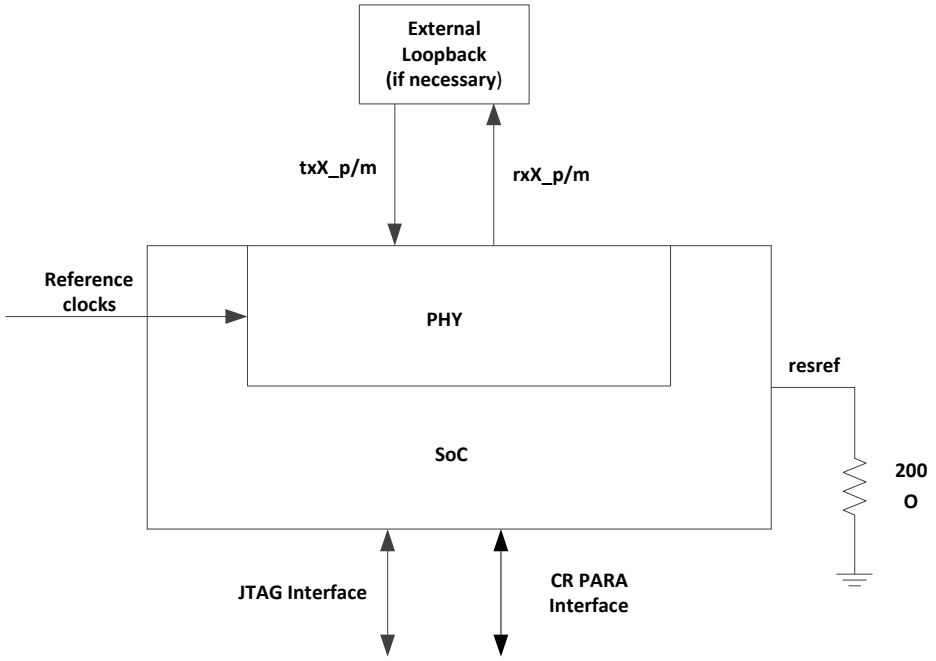


Table 2-2 shows the termination requirements for various tests.

Table 2-2 Termination Requirements

Test	Termination Required
JTAG_NO_CLOCK	None
JTAG_CLOCK	None
Simple Register Test	None
Register Test	None
BERT	External Loopback
INTERNAL LOOPBACK	None
LFPS SigDet	External Loopback
TX DC Levels	None
Common Mode	None
TX RX Detect	External Loopback

The ATE Verilog Testbench deliverables provide a pin list that lists the external requirements of the IP's top-level pins to place the IP in test mode. For this list, see Appendix A, "ATE Override Control Pin List".

Attention To ensure proper operation in silicon, all input pins apart from JTAG and Control Register (CR) Parallel ports, that are not listed in Appendix A should be randomized.

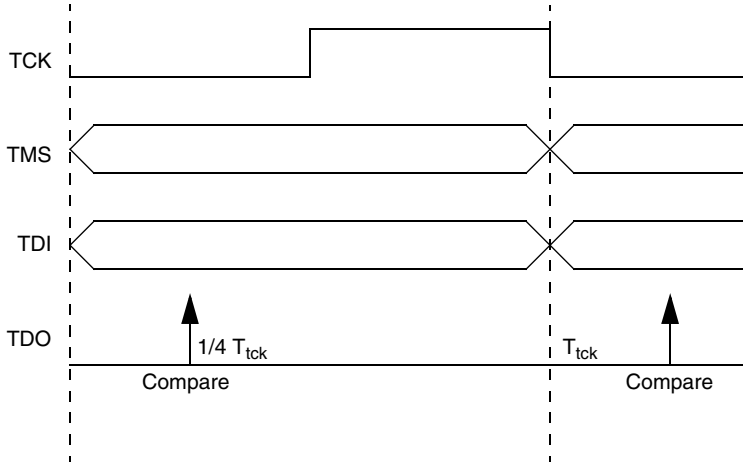
2.4 ATE Tests

This section describes the timing requirements and the ATE test suite supported by the SERDES_32G_PHY.

2.4.1 Timing

Timing for sampling and comparing TDO occurs before the rising edge of the TCK. We suggest ensuring that the TDO comparison with the expected TDO (as found in the generated ATE vector) occurs one quarter period after the falling edge of TCK, which helps ensure that TDO is stable at that point. Figure 2-2 shows a timing diagram of this relationship.

Figure 2-2 JTAG Timing Diagram



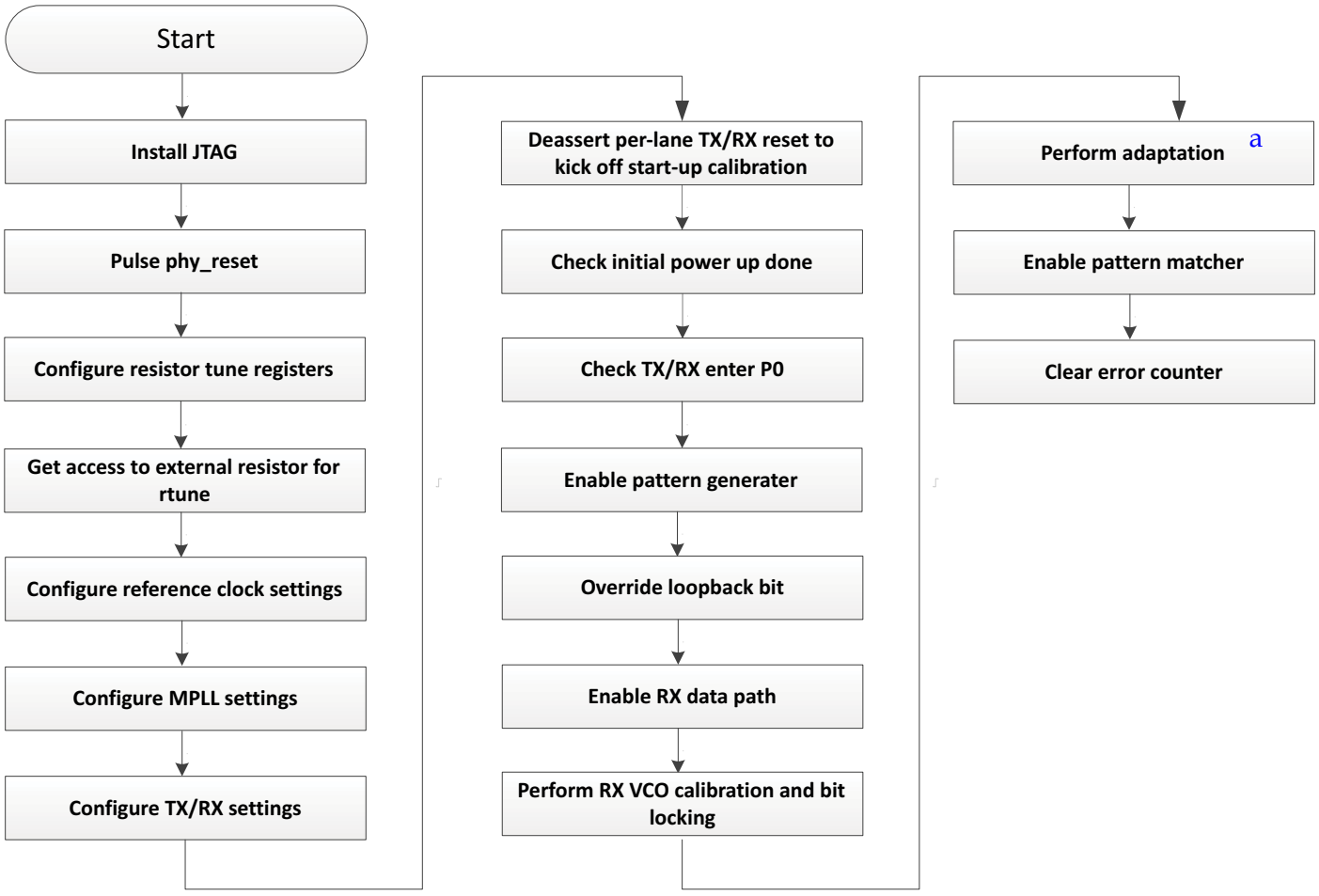
Note
1. Sampling TDO 1/4 of a cycle after a falling edge of TCK is recommended.
2. JTAG timing is dependent on the IO's of the ASIC in which the IP is implemented.

2.4.2 ATE Initialize Test Task

The ATE initialize test task is used to power up the PHY to P0 at the beginning of the tests. The raw-pcs firmware is enabled and rx adaptation can be performed when the baud rate >= 8 Gbps.

For tasks that run with raw-pcs firmware enabled, see Figure 2-3 ("ATE Initialization Sequence").

Figure 2-3 ATE Initialization Sequence



a–Tested only when the baud rate ≥ 8 Gbps, and external loopback is used (<PREFIX>_WAFER_SORT is undefined)

2.4.3 ATE Test Suite

Table 2-3 lists the ATE tests that are currently supported and a summary of coverage appears in Table 2-4.

Note that ATE sequence is designed for only a single PHY, hence it overrides res_ack_in to 1'b1 and res_req_in to 1'b0. If multiple PHYs sharing external resistors need to do simultaneous ATE testing, then res_req/ack should not be overridden by ATE vector, instead the functional connectivity described in "Reference Resistor Contention Bus" section of the PHY databook should be used.

Table 2-3 ATE Tests

Test	Page	Description
JTAG_NO_CLOCK	page 31	Read the JTAG id in the JTAG state matching register without reference clock (does not use CR)
JTAG_CLOCK	page 32	Read the JTAG id in the JTAG state matching register with reference clock
Simple Register Test	page 33	Read/write several registers in the PHY to verify register access interfaces
Register Test	page 34	Read default value of all readable, write/readable registers. Write the inverse and read it back. Write the default value back and read it back. Note: For ATE register test, the SHORT_RESET can't be defined. All other ATE tests can run in SHORT_RESET mode
BERT (1.5) BERT (2.5G) BERT (3G) BERT (5G) BERT (6G) BERT (8G) BERT (16G) BERT (25G) BERT (32G)	page 35	Using external loopback, generate and match LFSR. Verify no errors and send one error.
INTERNAL LOOPBACK (1.5G) INTERNAL LOOPBACK (2.5G) INTERNAL LOOPBACK (3G) INTERNAL LOOPBACK (5G) INTERNAL LOOPBACK (6G) INTERNAL LOOPBACK (8G) INTERNAL LOOPBACK (16G) INTERNAL LOOPBACK (25G) INTERNAL LOOPBACK (32G)	page 37	Using internal loopback, generate and match LFSR. Verify no errors and send one error.

Table 2-3 ATE Tests (Continued)

Test	Page	Description
LFPS SigDet	page 39	Check the functionality of SigDet circuitry using PHY's built in beaconing function
TX DC Levels	page 41	Check whether absolute maximum DC output high voltage (VOHmax) from the PHY is above a given level and that the absolute minimum DC output high voltage (VOHmin) is below a given level
Common Mode	page 44	Check common mode voltage of transmitter
TX RX Detect	page 47	Check receiver detection functionality of the PHY

Table 2-4 ATE Test Coverage

	ATE Tests									
Block	**JTAG_NO_CLOCK**	**JTAG_CLOCK**	**Simple Register Test**	**Register Test**	**BERT Tests**	**INTERNAL_LOOPBACK**	**LFPS SigDet**	**TX DC Levels**	**Common Mode**	**TX RX Detect**
Lane Receiver Analog Blocks										
Termination					X		X	X	X	X
CTLE					X	X	X	X	X	X
DFE+Slicer					X	X	X	X	X	X
Deserializer					X	X	X	X	X	X
CDR					X	X	X	X	X	X
Clock Path (VCO, DLLs, and Dividers)					X	X	X	X	X	X
RX Offset Cancellation					X	X	X	X	X	X
RX Slicer Calibration					X	X	X	X	X	X
RX Adaptation and Equalization[a]					X		X	X	X	X

Table 2-4 ATE Test Coverage (Continued)

	ATE Tests									
Block	**JTAG_NO_CLOCK**	**JTAG_CLOCK**	**Simple Register Test**	**Register Test**	**BERT Tests**	**INTERNAL_LOOPBACK**	**LFPS SigDet**	**TX DC Levels**	**Common Mode**	**TX RX Detect**
Lane Transmitter Analog Blocks										
Termination					X		X	X	X	X
LFPS							X			
Driver					X	X	X	X	X	X
RX Detection										X
Serializer					X	X	X	X	X	X
Loopback Path						X				
Clock Path and Dividers					X	X	X	X	X	X

Table 2-4 ATE Test Coverage (Continued)

	ATE Tests									
Block	**JTAG_NO_CLOCK**	**JTAG_CLOCK**	**Simple Register Test**	**Register Test**	**BERT Tests**	**INTERNAL_LOOPBACK**	**LFPS SigDet**	**TX DC Levels**	**Common Mode**	**TX RX Detect**
Sup Common/PLL Analog Block										
Bandgap/Biasing					X	X	X	X	X	X
Resistor Tuning					X		X	X	X	X
Prescaler					X	X	X	X	X	X
MPLL					X	X	X	X	X	X

a. Tested only when the baud rate ≥ 8 Gbps, and external loopback is used (<PREFIX>_WAFER_SORT is undefined)

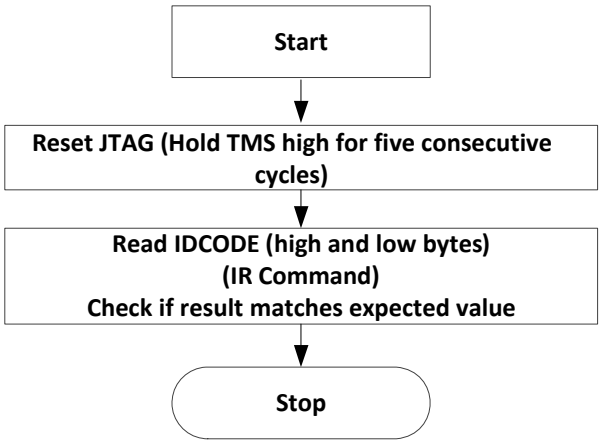
2.4.3.1 JTAG_NO_CLOCK Test

External Loopback Required: No
Wafer Sort: Yes

This test verifies basic functions of the JTAG state machine by sending an IR IDCODE command and reading the IDCODE. This test requires only that the power to the PHY is present. This test does not provide must-test coverage, but it verifies that the PHY can be accessed through the JTAG port. This is a good 'first test' to rule out connection issues from the tester to the PHY.

Test flow details are shown in Figure 2-4.

Figure 2-4 JTAG_NO_CLOCK Test



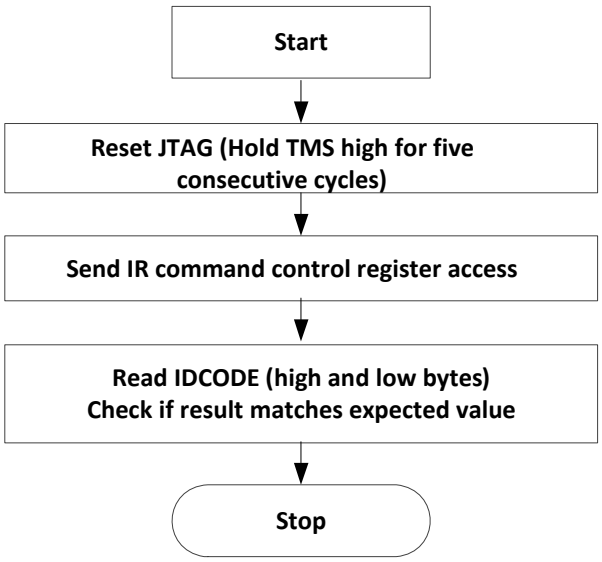
2.4.3.2 JTAG_CLOCK Test

External Loopback Required: No
Wafer Sort: Yes

This test is similar to the JTAG_NO_CLOCK test, but JTAG_CLOCK reads the JTAG ID value from the register in the PHY. To read registers in the PHY, a reference clock must be present, and the PHY must be enabled. After running JTAG_NO_CLOCK, JTAG_CLOCK is a good 'second test' to run to verify both the reference clock is present and the PHY is enabled.

Test flow details are shown in Figure 2-5.

Figure 2-5 JTAG_CLOCK Test



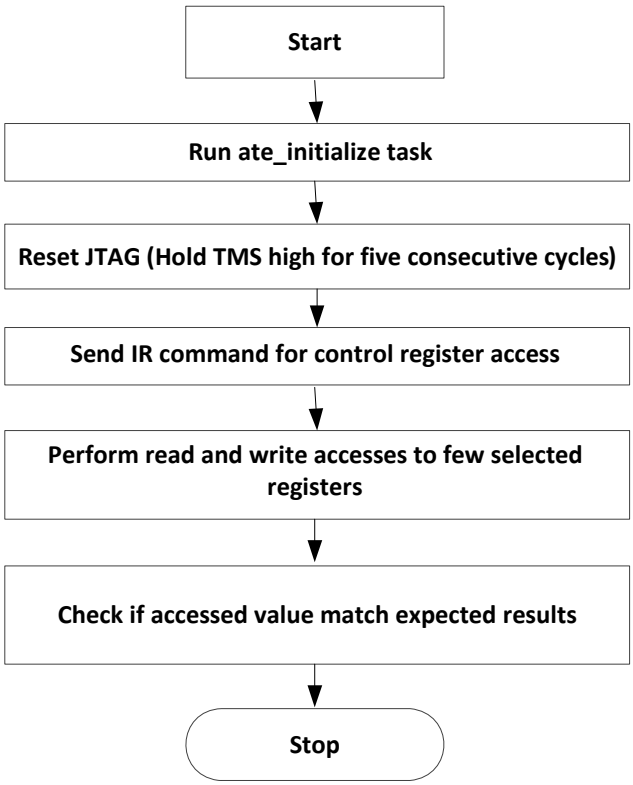
2.4.3.3 Simple Register Test

External Loopback Required: No
Wafer Sort: Yes

This test reads and writes several registers in the PHY to verify register access interfaces.

Test flow details are shown in Figure 2-6.

Figure 2-6 Simple Register Test



2.4.3.4 Register Test

External Loopback Required: No
Wafer Sort: Yes

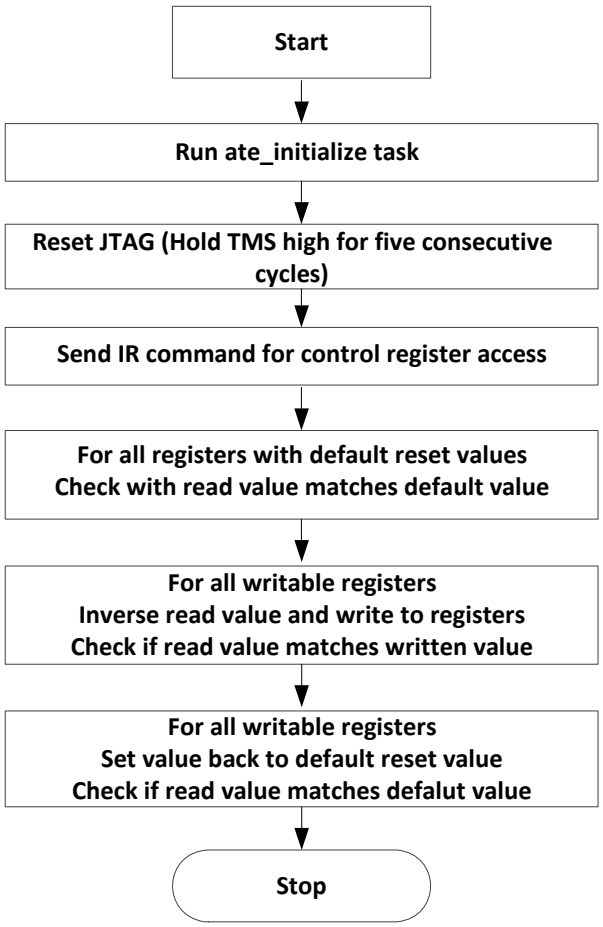
> **Note** <PREFIX>_SHORT_RESET must not be defined during this test to expedite the simulations. Defining <PREFIX>_SHORT_RESET may result in false vector failures.

Test Flow Summary

1. This test reads the default value of all readable registers in the PHY.
2. Then it writes the inverse pattern and verifies the result.
3. Finally, the test writes the default value back to the registers and verifies the result.

Details are shown in Figure 2-7.

Figure 2-7 Register Test Flow



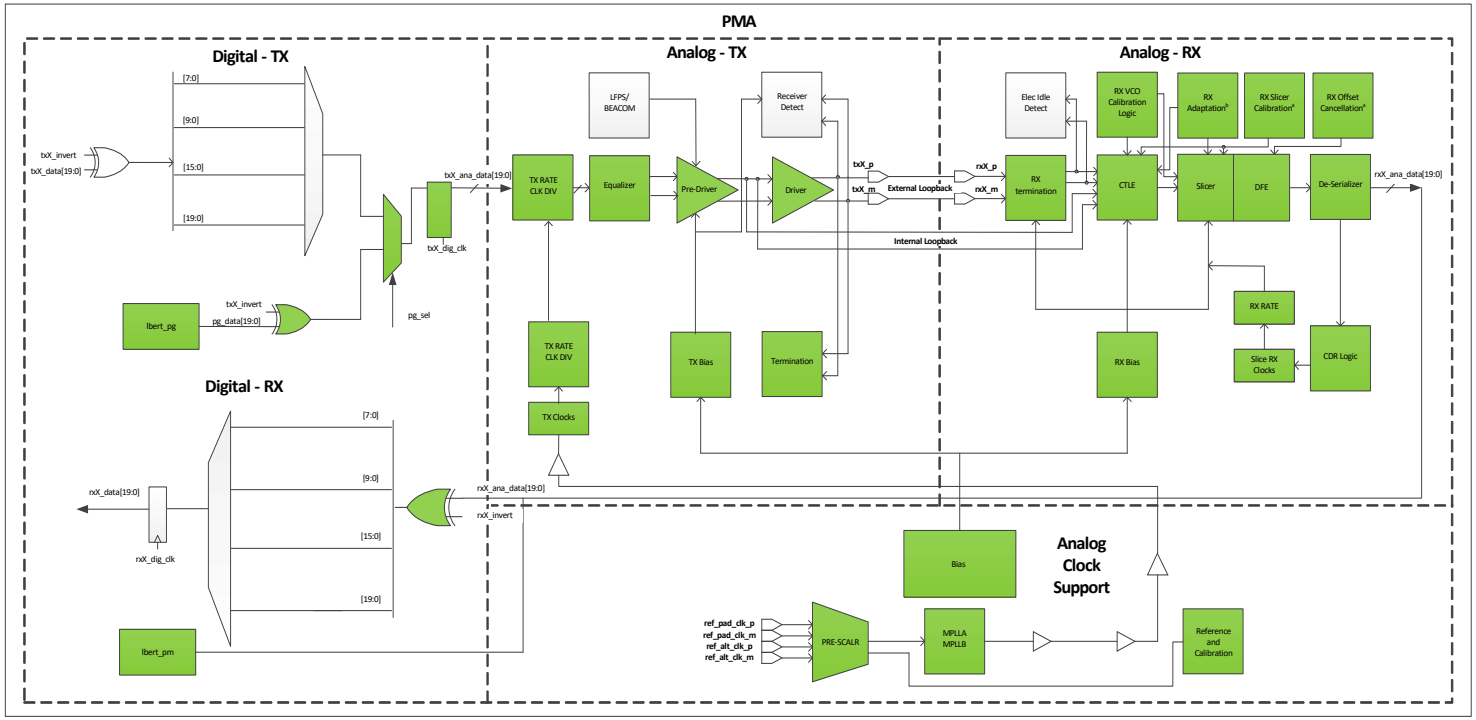
2.4.3.5 BERT Tests (1.5G, 2.5G, 3.5G, 5G, 6G, 8G, 16G, 25G, 32G)

External Loopback Required: Yes
Wafer Sort: No

This test exercises the following:

- RX/TX Data Path
- Error counters
- RX Slicer Calibration
- Pattern Matchers
- RX VCO calibration
- RX Offset Cancellation
- Pattern Generators
- TX/RX power state configuration
- MPLL and reference clock circuit
- RX Adaptation and Equalization

Figure 2-8 BERT Test Data Path

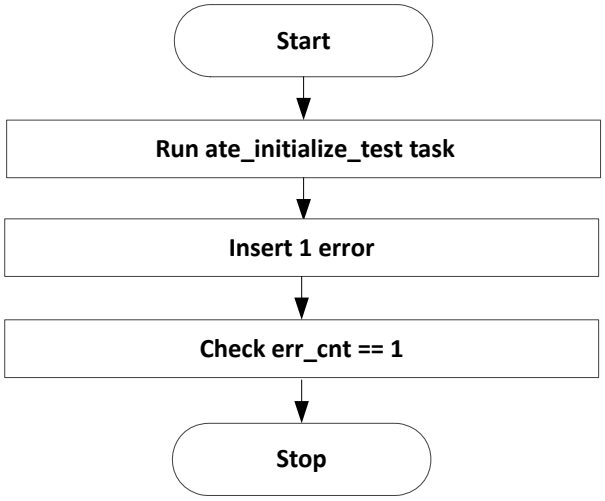


Test Flow Summary

1. This test initializes the transmitter and receiver into P0.
2. The lane pattern generator sends a known pattern.
3. The test matches the known pattern with the lane pattern matcher.

For test flow details, see Figure 2-9.

Figure 2-9 BERT Test Flow



2.4.3.6 Internal Loopback Tests (1.5G, 2.5G, 3G, 5G, 6G, 8G, 16G, 25G, 32G)

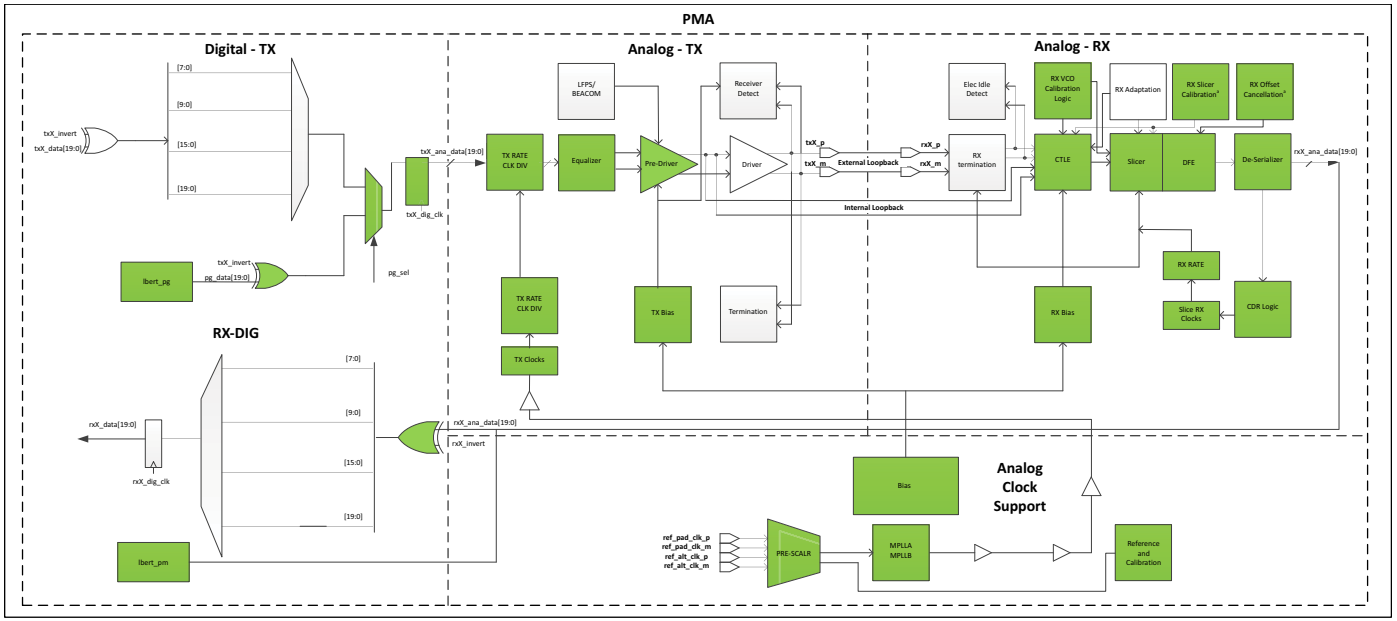
External Loopback Required: No
Wafer Sort: Yes

Note: <PREFIX>_WAFER_SORT must be defined during this test or unexpected test behavior may result.

This test exercises the following:

- RX/TX Data Path
- Error counters
- RX Slicer Calibration
- Pattern Matchers
- TX/RX power state configuration
- Pattern Generators
- Internal loopback path for TX/RX
- MPLL and reference clock circuit
- RX Offset Cancellation

Figure 2-10 Internal Loopback Test Data Path

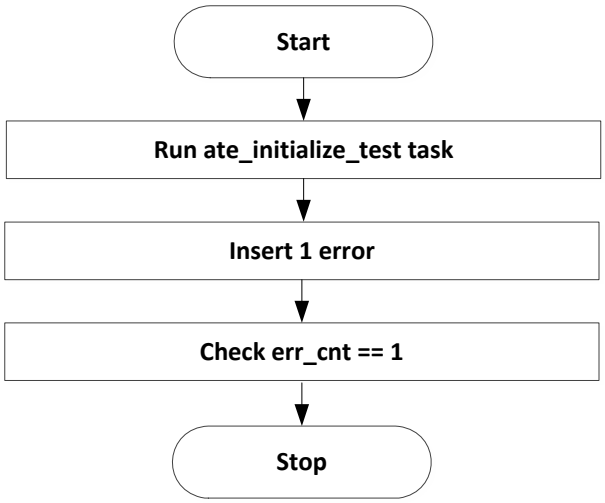


Test Flow Summary

1. This test initializes the transmitter and receiver into P0.
2. The lane pattern generator sends a known pattern.
3. The test matches the known pattern with the lane pattern matcher.

For test flow details, see Figure 2-11.

Figure 2-11 Internal Loopback Test Flow



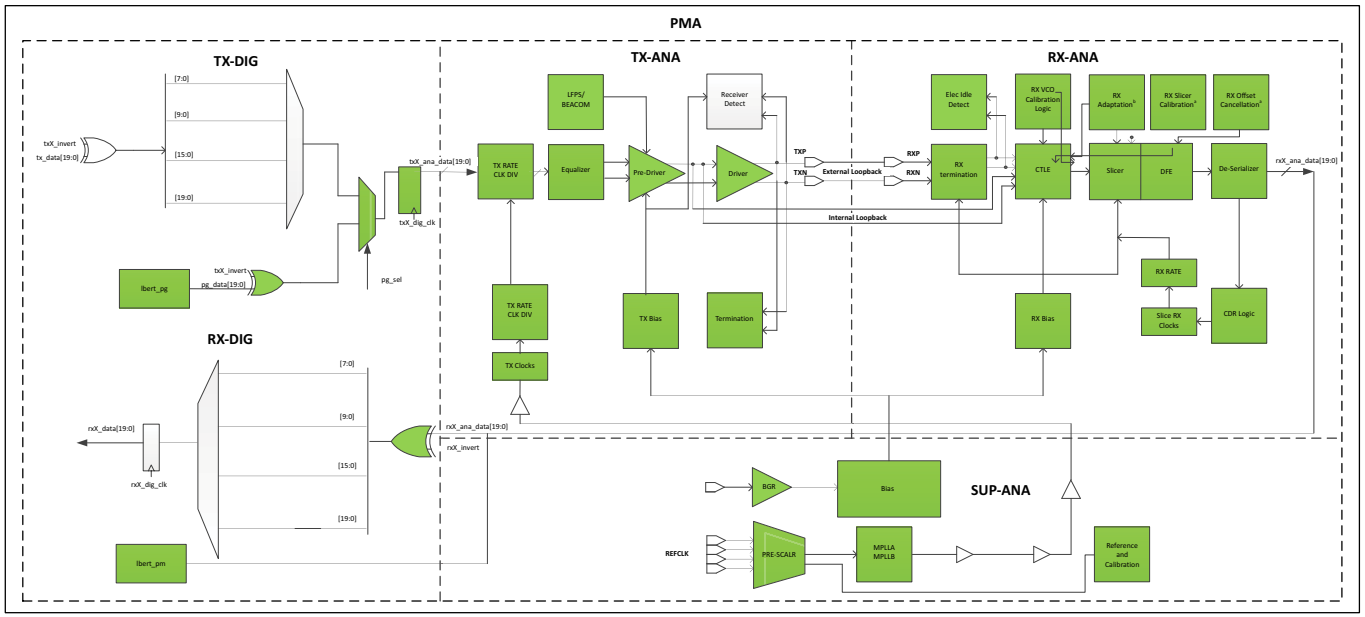
2.4.3.7 LFPS SigDet Test

External Loopback Required: Yes
Wafer Sort: No

This test exercises the following:

- RX/TX Data Path
- Error counters
- RX Slicer Calibration
- Pattern Matchers
- RX VCO calibration
- RX Offset Cancellation
- Pattern Generators
- TX/RX power state configuration
- MPLL and reference clock circuit
- LFPS
- SigDet
- RX Adaptation and Equalization

Figure 2-12 LFPS SigDet Test

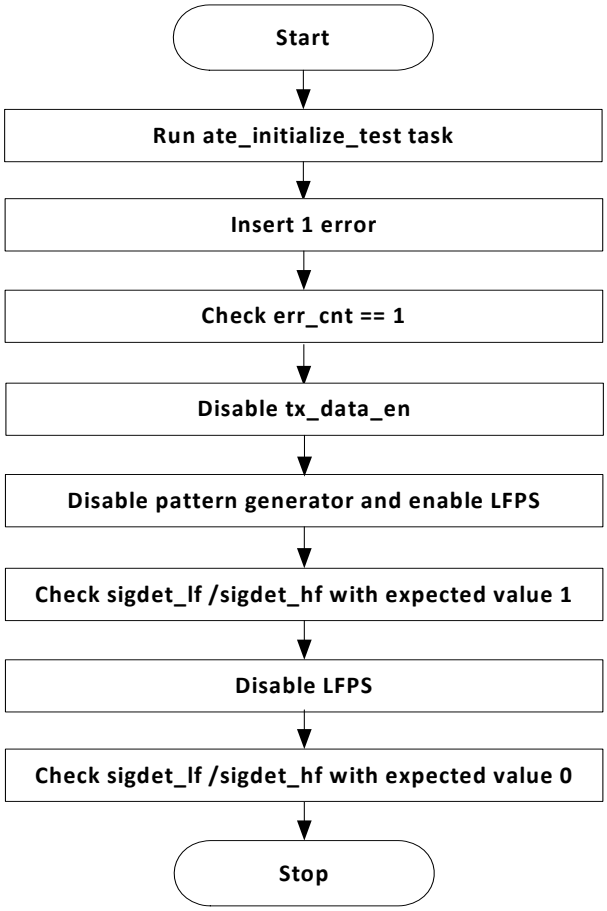


Test Flow Summary

1. This test initializes the transmitter and receiver into P0.
2. The lane pattern generator sends a known pattern.
3. The test matches the known pattern with the lane pattern matcher.
4. The pattern generator is disabled and LFPS is enabled.
5. The sigdet_lf or sigdet_hf values are read back and checked against an expected value of 1.
6. The LFPS is disabled.
7. The sigdet_lf or sigdet_hf values are read back and checked against an expected value of 0.

For test flow details, see Figure 2-13.

Figure 2-13 LFPS SigDet Test Flow



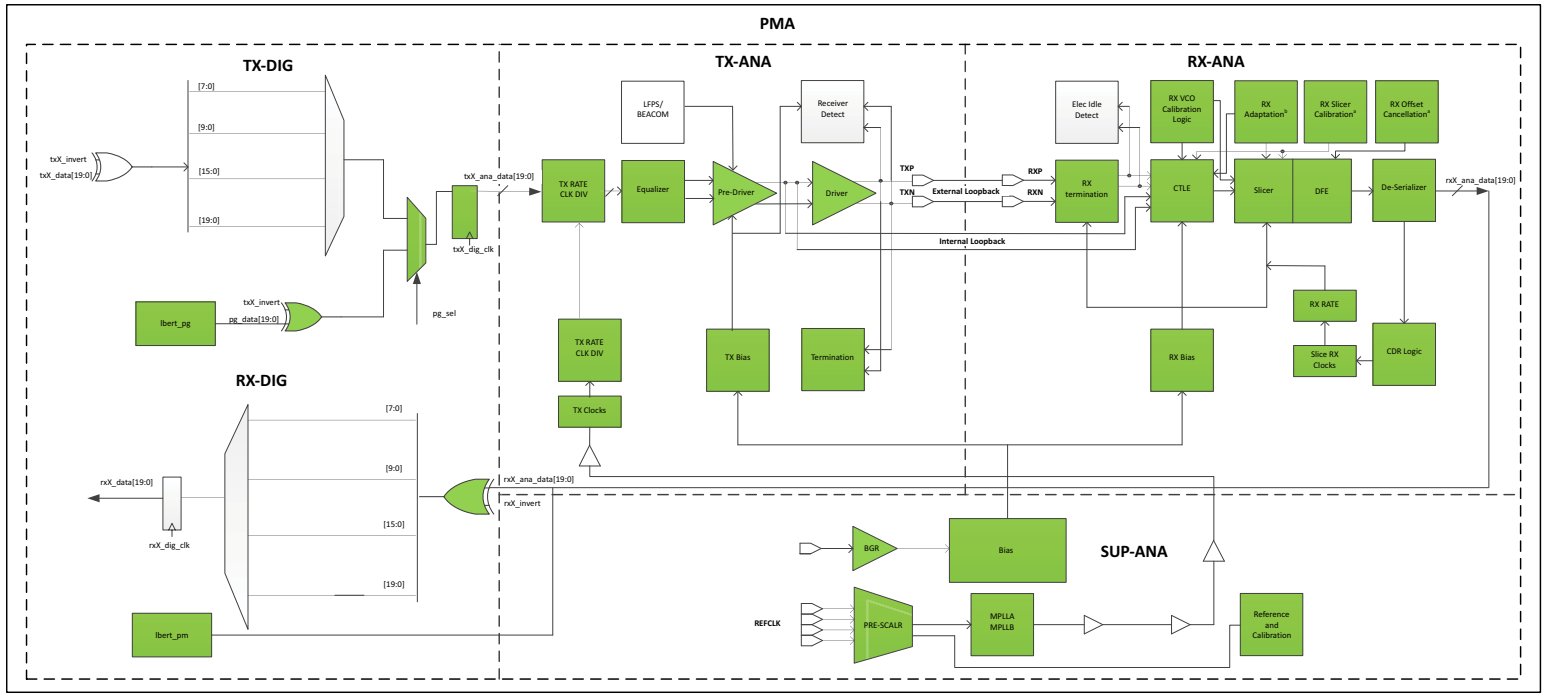
2.4.3.8 TX DC Levels Test

External Loopback Required: No
Wafer Sort: Yes

This test exercises the following:

- RX/TX Data Path
- Error counters
- RX Slicer Calibration
- Pattern Matchers
- RX VCO calibration
- RX Offset Cancellation
- Pattern Generators
- TX/RX power state configuration
- MPLL and reference clock circuit
- RX Adaptation and Equalization

Figure 2-14 TX DC Levels Test



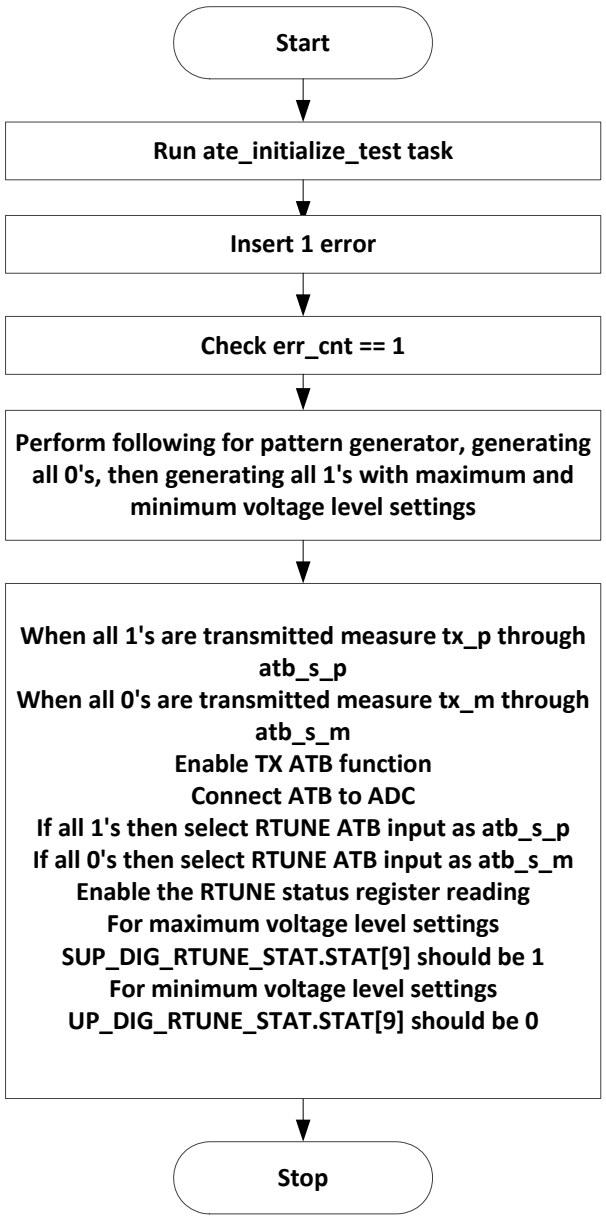
Test Flow Summary

This test:

1. Initializes the transmitter and receiver into P0.
2. Sends a known pattern from the lane pattern generator.
3. The test matches the known pattern with the lane pattern matcher.
4. Sends an “all 1's” pattern from the pattern generator.
5. Sets the voltage level to the maximum.
6. Measures voltage on txX_p through atb_s_p.
7. Enables the TX ATB function.
8. Selects the RTUNE ATB input as atb_s_p.
9. Enables reading the RTUNE status register.
10. Reads SUP_DIG_RTUNE_STAT.STAT[9] and checks against an expected value of 1.
11. Sets the voltage level to the minimum.
12. Measures voltage on txX_p through atb_s_p.
13. Enables the TX ATB function.
14. Selects the RTUNE ATB input as atb_s_p.
15. Enables reading the RTUNE status register.
16. Reads SUP_DIG_RTUNE_STAT.STAT[9] and checks against an expected value of 0.
17. Transmits an “all 0's” pattern from the pattern generator.
18. Sets the voltage level to the maximum.
19. Measures voltage on txX_m through atb_s_m.
20. Enables the TX ATB function.
21. Selects the RTUNE ATB input as atb_s_m.
22. Enables reading the RTUNE status register.
23. Reads SUP_DIG_RTUNE_STAT.STAT[9] and checks against an expected value of 1.
24. Sets the voltage level to the minimum.
25. Measures voltage on txX_m through atb_s_m.
26. Enables the TX ATB function.
27. Selects the RTUNE ATB input as atb_s_m.
28. Enables reading the RTUNE status register.
29. Reads SUP_DIG_RTUNE_STAT.STAT[9] and checks against an expected value of 0.

For test flow details, see Figure 2-15.

Figure 2-15 TX DC Levels Test Flow



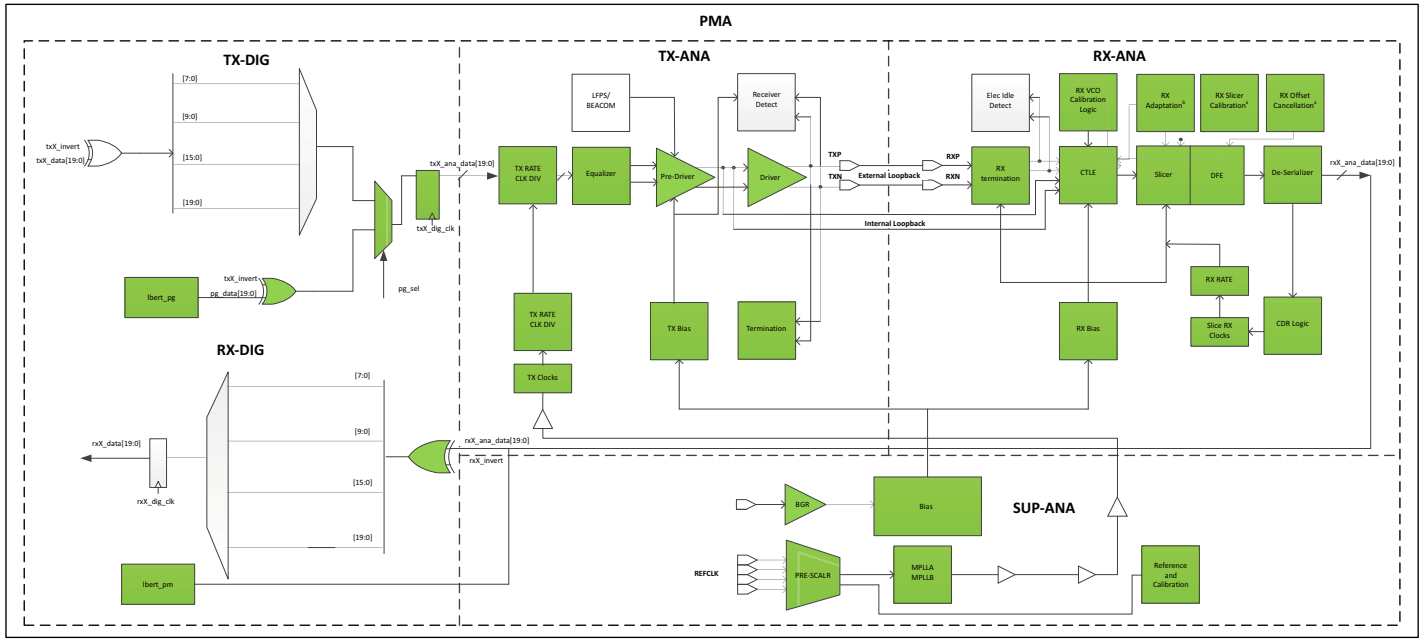
2.4.3.9 Common Mode Test

External Loopback Required: No
Wafer Sort: Yes

This test exercises the following:

- RX/TX Data Path
- Error counters
- RX Slicer Calibration
- Pattern Matchers
- RX VCO calibration
- RX Offset Cancellation
- Pattern Generators
- TX/RX power state configuration
- MPLL and reference clock circuit
- RX Adaptation and Equalization

Figure 2-16 Common Mode Test



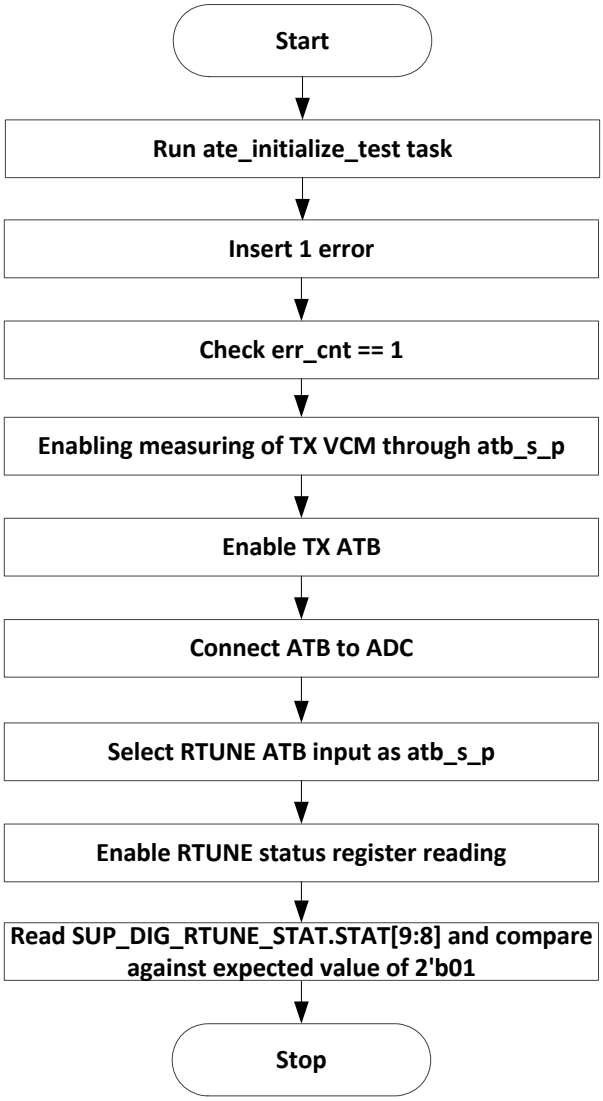
Test Flow Summary

This test:

1. Initializes the transmitter and receiver into P0.
2. Sends a known pattern from the lane pattern generator.
3. The test matches the known pattern with the lane pattern matcher.
4. Enables measuring of TX VCM through atb_s_p.
5. Enables TX ATB.
6. Connects ATB to ADC.
7. Selects the RTUNE ATB input as atb_s_p.
8. Enables reading the RTUNE status register.
9. Reads SUP_DIG_RTUNE_STAT.STAT[9:8] and checks against an expected value of 2'b01.

For test flow details, see Figure 2-17 on page 46.

Figure 2-17 Common Mode Test Flow



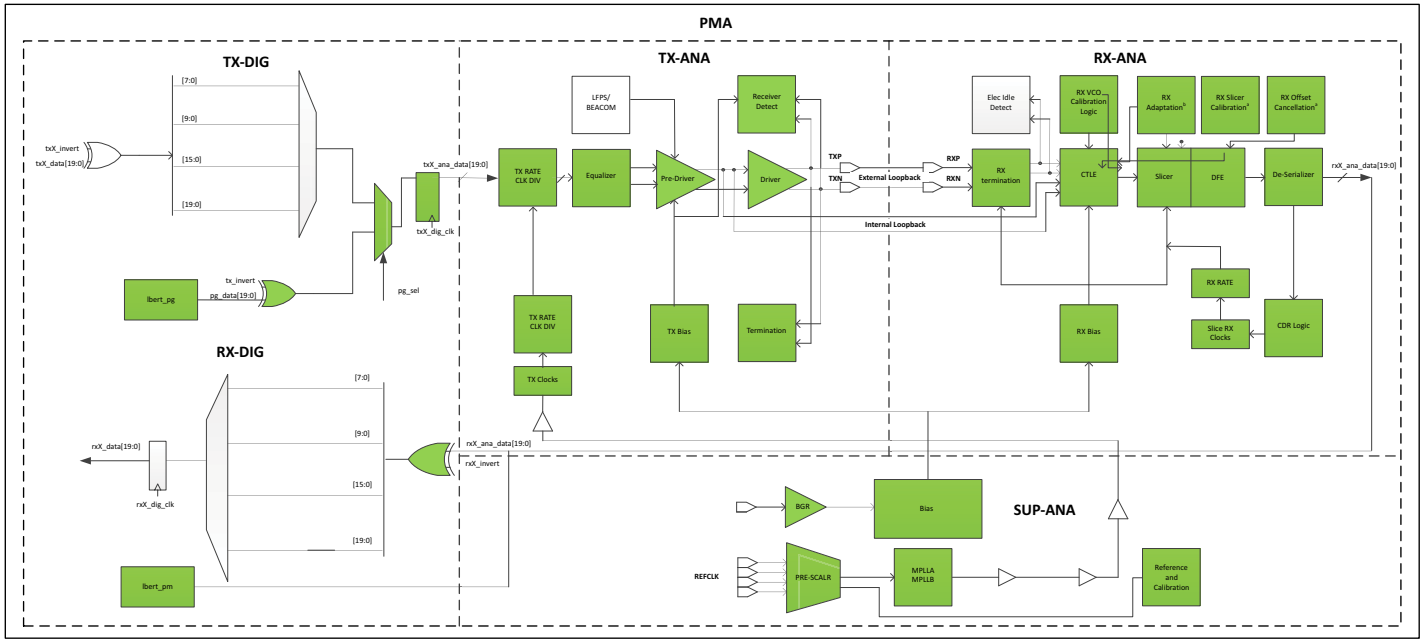
2.4.3.10 TX RX Detect Test

External Loopback Required: Yes
Wafer Sort: No

This test exercises the following:

- RX/TX Data Path
- Error counters
- RX Slicer Calibration
- Pattern Matchers
- RX VCO calibration
- RX Offset Cancellation
- Pattern Generators
- TX/RX power state configuration
- RX Adaptation and Equalization
- MPLL and reference clock circuit
- Receiver Detect

Figure 2-18 TX RX Detect Test



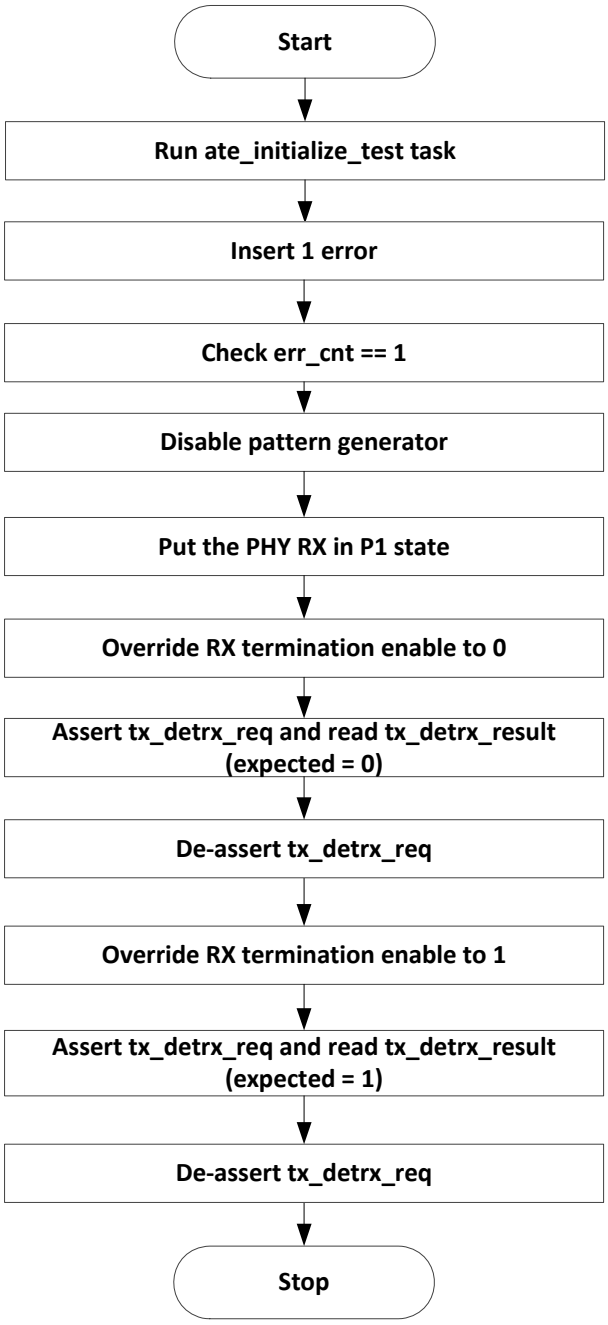
Test Flow Summary

This test:

1. Initializes the transmitter and receiver into P0.
2. Sends a known pattern from the lane pattern generator.
3. The test matches the known pattern with the lane pattern matcher.
4. Disables the pattern generator.
5. Puts the PHY RX in the P1 state.
6. Overrides the RX termination enable to 0.
7. Asserts tx_detrx_req and read tx_detrx_result (expected = 0).
8. De-asserts tx_detrx_req.
9. Overrides RX termination enable to 1.
10. Asserts tx_detrx_req and read tx_detrx_result (expected = 1).
11. De-asserts tx_detrx_req.

For test flow details, see Figure 2-19 on page 49.

Figure 2-19 TX RX Detect Test Flow



A
ATE Override Control Pin List

Table A-1 contains the ATE override control pin list for the SERDES_32G_PHY at the PHY and PIPE levels. Connect the signals listed in Table A-1 with a multiplex and an override register that are controllable during ATE. To help understand the table information, note the following about controlling the raw-PCS firmware FSM and RX adaptation:

- RX adaptation is explicitly disabled when the <PREFIX>_WAFER_SORT macro is defined.
- When <PREFIX>_WAFER_SORT is undefined, RX adaptation is automatically enabled. Under these conditions, you can explicitly disable RX adaptation by setting the following in the ate_param_settings.v file:

```
ate_adapt_reqd=1'b0;
```

Table A-1 ATE Override Control Pin List for SERDES_32G_PHY and PIPE

PHY		PIPE	
Inputs	**Default Value**	**Inputs**	**Default Value**
pg_mode_en	0	pg_mode_en	0
phy_reset	Pulse to subject the DUT to POR sequence	phy_reset	Pulse to subject the DUT to POR sequence
ref_alt0_clk_m	Based on configuration	phyN_ref_alt0_clk_m	Based on configuration
ref_alt0_clk_p	Based on configuration	phyN_ref_alt0_clk_p	Based on configuration
ref_alt1_clk_m	Based on configuration	phyN_ref_alt1_clk_m	Based on configuration
ref_alt1_clk_p	Based on configuration	phyN_ref_alt1_clk_p	Based on configuration
ref_pad_clk_m	Based on configuration	phyN_ref_pad_clk_m	Based on configuration
ref_pad_clk_p	Based on configuration	phyN_ref_pad_clk_p	Based on configuration
refa_clk_en	Based on configuration	phy_laneN_refa_clk_en	Based on configuration
		ext_pclk_req	1
refb_clk_en	Based on configuration	phy_laneN_refb_clk_en	Based on configuration

Table A-1 ATE Override Control Pin List for SERDES_32G_PHY and PIPE (Continued)

PHY		PIPE	
Inputs	**Default Value**	**Inputs**	**Default Value**
refa_clk_sel	Based on configuration	phyN_refa_clk_sel	Based on configuration
refb_clk_sel	Based on configuration	phyN_refb_clk_sel	Based on configuration
refa_range	Based on configuration	phyN_refa_range	Based on configuration
refb_range	Based on configuration	phyN_refb_range	Based on configuration
refa_clk_div2_en	Based on configuration	phyN_refa_clk_div2_en	Based on configuration
refb_clk_div2_en	Based on configuration	phyN_refb_clk_div2_en	Based on configuration
refa_raw_clk_div2_en	Based on configuration	phyN_refa_raw_clk_div2_en	Based on configuration
refb_raw_clk_div2_en	Based on configuration	phyN_refa_raw_clk_div2_en	Based on configuration
bs_ce	0	phyN_bs_ce	0
scan_mode	0	phyN_scan_mode	0
scan_pma_occ_en	0	phyN_scan_pma_occ_en	0
scan_shift	0	phyN_scan_shift	0
scan_shift_cg	0	phyN_scan_shift_cg	0
test_burnin	0	phy_test_burnin	0
test_powerdown	0	phy_test_powerdown	0
rxX_reset	1		
txX_reset	1		
test_flyover_en	0	phyN_test_flyover_en	0
test_stop_clk_en	0	phy_test_stop_clk_en	0
sram_bypass	0	phyN_sram_bypass	0
sram_bootload_bypass	Based on configuration	phyN_sram_bootload_bypass	Based on configuration
sram_ext_ld_done	0	phyN_sram_ext_ld_done	0
nominal_vph_sel	Based on configuration	phyN_nominal_vph_sel	Based on configuration
nominal_vp_sel	Based on configuration	phyN_nominal_vp_sel	Based on configuration
laneX_cntx_en	1	phy_laneX_cntx_en	1
cr_para_sel	Based on configuration	phyN_cr_para_sel	Based on configuration
		pcs_scan_mode	0

Table A-1 ATE Override Control Pin List for SERDES_32G_PHY and PIPE (Continued)

PHY		PIPE	
Inputs	**Default Value**	**Inputs**	**Default Value**
		pcs_scan_shift	0
		pcs_scan_shift_cg	0
		phy_ext_ctrl_sel	Based on configuration
		pipe_laneX_reset_n	0
		pipe_laneX_lanepll_bypass_mode	5'b11111
		pipe_laneX_protocol	Based on configuration
rst_sms (only applicable for internal SRAM customization projects, refer to PHY databook)	~phy_reset	phy0_rst_sms (only applicable for internal SRAM customization projects, refer to PHY databook)	~phy_reset
sram_int_access_en (only applicable for internal SRAM customization projects, refer to PHY databook)	0	phy0_sram_int_access_en (only applicable for internal SRAM customization projects, refer to PHY databook)	0
ana_pwr_en (only applicable for power gating with LDO switch customization projects, refer to PHY databook)	1	phy0_ana_pwr_en (only applicable for power gating with LDO switch customization projects, refer to PHY databook)	1

Note

- For sram_bootload_bypass, it needs to be set to 1 to bypass the SRAM bootload if only SRAM is used. If external ROM or internal ROM exists, it needs to be set to 0.
- The CR parallel or JTAG interface needs to be selected for the ATE test. For the CR parallel and JTAG interface details, refer to section 1.3.3 “JTAG Accessibility” and 1.3.4 “Control Register (CR) Parallel Accessibility”