Xena ANLT Utility Documentation

Release 2.2.3

Teledyne LeCroy Xena

Mar 22, 2024

TABLE OF CONTENT

Xena ANLT Utility (*Xena ANLT Utility*) provides an **interactive shell** for users to control Xena testers via **command-line interface** (CLI) commands to performance operations such as *ANLT* testing and resource management (more will be added in future releases) in an interactive fashion.

CHAPTER

ONE

INTRODUCTION

Xena OpenAutomation ANLT Utility (*Xena ANLT Utility*) is an application that provides users with a command-line user interface to do interactive testing (e.g. ANLT testing). It provides a set of CLI commands to manage and configure test ports, collect statistics, and save logs. Xena ANLT Utility uses xoa-driver to communicate to Xena ANLT Testers.

Fig. 1: *Xena ANLT Utility System*

The CLI commands of Xena ANLT Utility are categorized into the following families:

- *Auto-Negotiation and Link Training*
  - *AN Functionalities*
  - *LT Functionalities*
- *Test Resource Management*

See also:

You can view a list of Xena ANLT Utility CLI commands in *Summary of Xena ANLT Utility CLI commands*

1.1 Auto-Negotiation and Link Training

Auto-Negotiation and Link Training (*ANLT*) provides functions to help you fine-tune the protocol to its optimal state, test interoperability between different vendors, and protocol compliance for different implementations.

Auto-negotiation (*AN*) was originally designed for Ethernet over twisted pair up to 1G. Beyond exchanging speed capabilities for the link participants, AN has evolved for today's Ethernet to include additional configuration information for establishing reliable and consistent connections. AN allows the devices at the end points of a link to negotiate common transmission parameters capabilities like speed and duplex mode, exchange extended page information and media signaling support. At higher speeds and signaling the choice of FEC may be relevant. It is during auto negotiation the end points of a link share their capabilities and choose the highest performance transmission mode they both support.

Once the ports in the link have completed the requisite AN information exchange and reached agreement, the link partners move to the next step, link training (LT), the exchange of Training Sequences. This is essential to tune the channels for optimal transmission. During link training the two end points of the link will exchange signals.

Summary

Management Command Summary #

Command	Description	Example
connect	Connect to tester	`connect 10.10.10.10 yourname`
port	Reserve and switch port	`port 0/0 | port 0/0 --reset`
ports	List ports	`ports | ports --all`
module-config	Set module media and port config	`module-config 0 osfp800 8 100g`
exit	Exit the session	`exit`

AN/LT Command Summary

Command	Description	Example
anlt start	Apply and start AN/LT on the port	`anlt start`
anlt stop	Stop AN/LT on the port	`anlt stop`
anlt log	Show AN/LT protocol trace log and save to a file	`anlt log --filename mylog.log`
anlt autorestart	Control AN/LT autorestart	`anlt autorestart --link-down --lt-fail`
anlt status	Show AN/LT status of the local port	`anlt status`
anlt strict	Enable/disable ANLt strict mode	`anlt strict --on`
anlt logctrl	Control what should be logged in ANLT by xenaserver	`anlt logctrl -DALGPNmTcsZO`

AN Command Summary

Command	Description	Example
an config	Configure AN of the local port	`an config --on --loopback`
an status	Show AN status of the local port	`an status`

LT Command Summary

Command	Description	Example
lt alg	Set the link training algorithm for the specified serdes	`lt alg 0 alg0`
lt config	Configure LT of the local port	`lt config --on --mode auto --preset0 standard --timeout enable`
lt dec	Request **remote port** to decrease (-) its emphasis value by 1	`lt dec 0 main`
lt no-eq	Request **remote port** to turn off equalizer on its emphasis	`lt dec 0 main`

Fig. 2: *Command Summary*

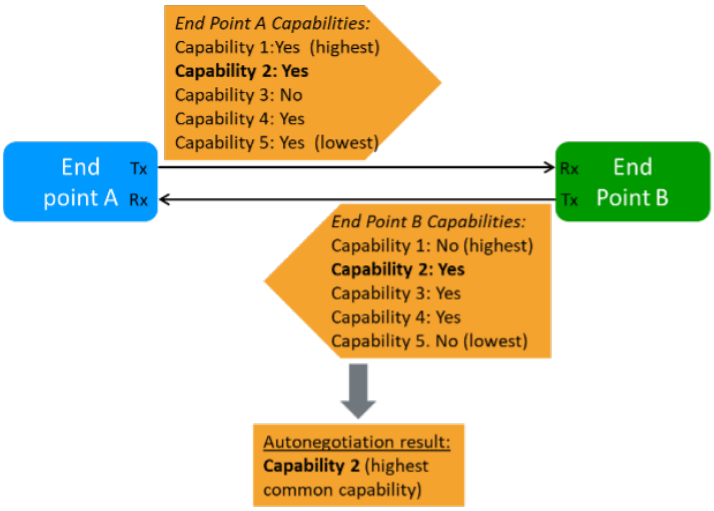


Fig. 3: Auto-Negotiation Process

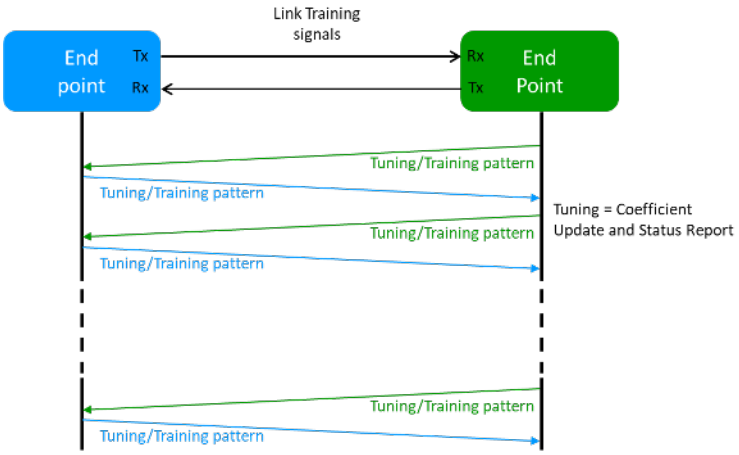


Fig. 4: Link Training Process

No Auto Negotiation, No Link Training

In some instances, Auto Negotiation and Link Training are not required to establish a communication path: High speed optical transceivers and interfaces typically only run at one speed, so there is no need the negotiate this.

Link Training is only required for electrical interfaces - in some cases (e.g. when short cables are used) an electrical interface may become operational just using default settings of the terminal equipment in the communication path. The IEEE 802.3 by specification allows for force connect over electrical interfaces in these instances.

No Auto Negotiation, Link Training

While Link Training can be essential to make some electrical interfaces work, Auto Negotiation may not be required, if the link speed is fixed or if it can be manually set at both end points of a link.

Auto Negotiation and Link Training

Auto Negotiation and Link Training are in principle two **independent** processes. However, when both are to be done, Auto Negotiation must start first to determine the overall mode for a link and then the Link Training. Hereby you get the sequence shown in the figure below.

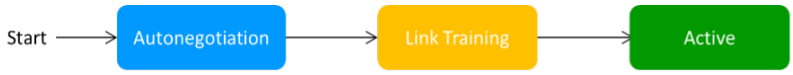


Fig. 5: Auto-Negotiation and Link Training Sequence

See also:

Read more about Auto Negotiation and Link Training on NRZ and PAM4 based Ethernet Interfaces.

In Xena ANLT Utility, you can find the following functionalities to do auto-negotiation and link training interactive tests.

1.1.1 AN Functionalities

1. Enable/disable auto-negotiation
2. Auto-negotiation trace log, provides AN trace log for debugging and troubleshooting.
3. Auto-negotiation status, provides the following AN status:
    - Received and transmitted number of Link Code Words (Base Pages), message pages, and unformatted pages
    - Number of HCD (Highest Common Denominator) failures
    - Number of FEC failures
    - Number of LOS (Loss of Sync) failures
    - Number of timeouts
    - Number of successes
    - Duration of AN in microseconds

1.1.2 LT Functionalities

1. Enable/disable link training
2. Allow/deny link training loopback
3. Enable/disable link training timeout
4. Tuning link partner TX EQ coefficient, use presets as a starting point to tune link partner TX EQ coefficients per serdes, increment and decrement of coefficients c(-3), c(-2), c(-1), c(0), c(1).
5. Configure local TX EQ coefficients
6. Monitor local TX EQ coefficients
7. Link training trace log per serdes
8. Link training status per serdes, provides the following LT status:
    - Number of lost locks
    - Local value of coefficient (per coefficient)
    - RX number of increment/decrement requests from link partner (per coefficient)
    - RX number of EQ coefficient request limits reached from link partner (per coefficient)
    - RX number of EQ request limits reached from link partner (per coefficient)
    - RX number of coefficients not supported from link partner (per coefficient)
    - RX number of coefficients at limit from link partner (per coefficient)
    - TX number of increment/decrement requests to link partner (per coefficient)
    - TX number of EQ coefficient request limits reached to link partner (per coefficient)
    - TX number of EQ request limits reached to link partner (per coefficient)
    - TX number of coefficients not supported to link partner (per coefficient)
    - TX number of coefficients at limit to link partner (per coefficient)
    - Duration of LT in microseconds
    - PRBS total error bits
    - PRBS total error bits
    - PRBS bit error rate
    - Local frame lock status
    - Link partner frame lock status

1.2 Test Resource Management

1. Connect to tester
2. Reserve port
3. Release port
4. Reset port
5. Disconnect

CHAPTER

TWO

GETTING STARTED

2.1 Installation

ANLT Utility is made into a Python package, xoa-utils for easy installation and upgrade for all platforms. **However, it requires you to have knowledge about Python and the operating system you are using**.

The table below shows the distribution methods for each platform.

Table 1: ANLT Utility Distribution

Windows (x64)	macOS	Linux
Python package xoa-utils (requires Python >=3.8) xoa-utils-win-x64-x.y.z.exe (64-bit, no installation required)	Python package xoa-utils (requires Python >=3.8)	Python package xoa-utils (requires Python >=3.8)

If you want to use Windows executable:

2.1.1 Executable for Windows

Generate SSH Key

ANLT Utility requires an SSH key pair to run as a SSH service. To generate a SSH key pair, please open Command Prompt or PowerShell on Windows.

```
> ssh-keygen -t rsa
```

You will be prompted to save and name the key. **If not found**, read Generate SSH Keys in Windows 10/11.

```
> Generating public/private rsa key pair. Enter file in which to save the key (C:\
↪Users\USER\.ssh\id_rsa):
```

Press `Enter` to use the default name `id_rsa`.

Important: The filename of the key should be `id_rsa`. Please don't use other filenames otherwise the application won't be able to run.

Next you will be asked to create and confirm a passphrase for the key:

```
> Enter passphrase (empty for no passphrase):
```

Press `Enter` to skip passphrase.

```
> Enter same passphrase again:
```

Press `Enter` again to confirm passphrase.

This will generate two files, by default called `id_rsa` and `id_rsa.pub` in `C:\Users\USER\.ssh`

See also:

You can read more about Generating SSH Key

Attention: If your machine doesn't have internet access, you should generate the SSK keys on another machine and copy the keys to your target machine.

Download Windows Executable

Download `xoa-utils-win-x64-`.

Start ANLT Utility

Unzip the file and run `xoa-utils-win-x64-<version>.exe`. The executable includes Python itself, xoa-driver, and all the dependencies.

Important: There is **no need to install Python or any Python packages** on your PC to run the ANLT Utility Windows executable, but remember **you still need to generate the SSH key**.

```
> xoa-utils
(PID: 12345) ANLT Utility SSH Service (1.1.0) running on 0.0.0.0:22622.
```

Note: Unlike the Python package, you can't change the port number on which you run the SSH server if using the Windows executable.

If you want to use Python package:

2.1.2 Python Package for macOS/Linux/Windows

Generate SSH Key

ANLT Utility requires an SSH key pair to run as a SSH service. To generate a SSH key pair, please open Command Prompt/PowerShell (Windows) or Terminal (macOS/Linux)

```
> ssh-keygen -t rsa
```

You will be prompted to save and name the key.

```
> Generating public/private rsa key pair. Enter file in which to save the key (/Users/
↪USER/.ssh/id_rsa):
```

Press `Enter` to use the default name `id_rsa`.

Important: The filename of the key should be `id_rsa`. Please don't use other filenames otherwise the application won't be able to run.

Next you will be asked to create and confirm a passphrase for the key:

```
> Enter passphrase (empty for no passphrase):
```

Press `Enter` to skip passphrase.

```
> Enter same passphrase again:
```

Press `Enter` again to confirm passphrase.

This will generate two files, by default called `id_rsa` and `id_rsa.pub` in `/Users/USER/.ssh`

See also:

You can read more about Generating SSH Key

Python Package for Win/macOS/Linux

Note: Skip this section, if you are a Windows (x64) user and don't want to install ANLT Utility as a Python package but simply want the **.exe** application,

---

ANLT Utility is available to install via the Python Package Index. You can also install from the source file. The steps below will guide you through

Prerequisites

Before installing ANLT Utility, please make sure your environment has installed:

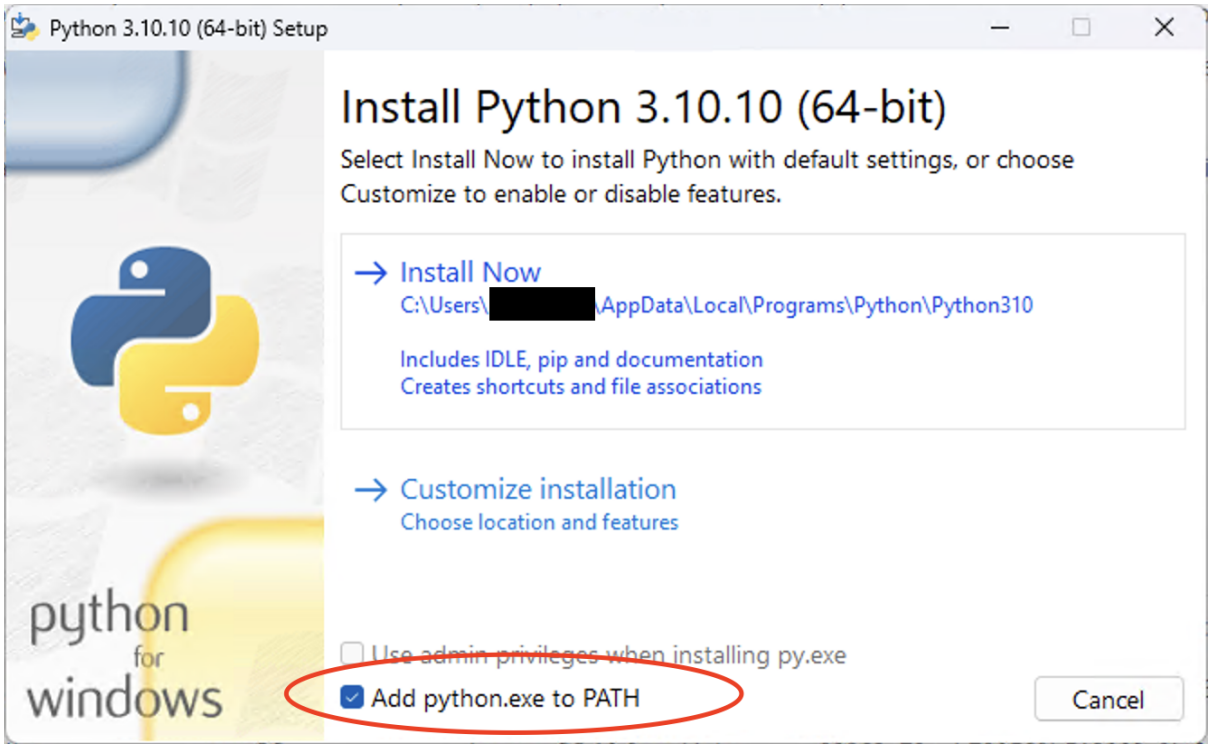
- *Install Python* (requires **Python >= 3.8**)
- *Install PIP*

Install Python

Important: ANLT Utility requires Python >= 3.8.

ANLT Utility requires that you download and install Python3 on your system.

Note: If you use **Windows**, remember to check **Add python.exe to PATH**.



After installation, open **Command Prompt** (Windows) or **Terminal** (macOS/Linux) and type `python` to verify your Python installation.

Listing 1: Check Python installation in Windows.

```
> python
Python 3.10.10 (tags/v3.10.10:878ead1, Feb  7 2023, 16:38:35) [MSC v.1934 64 bit␣
↪(AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Listing 2: Check Python installation in macOS/Linux.

```
$ python3
Python 3.10.10 (v3.10.10:a58ebcc701, Feb 7 2023, 14:50:16) [Clang 13.0.0 (clang-1300.
↪0.29.30)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Note: If you are stuck with Python installation, seek help in Python 3 Installation & Setup Guide

Install PIP

Make sure `pip` is installed on your system. `pip` is the package installer for Python . You can use it to install packages from the Python Package Index and other indexes.

Usually, `pip` is automatically installed if you are:

- working in a virtual Python environment (virtualenv or venv). It is not necessary to use `sudo pip` inside a virtual Python environment.
- using Python downloaded from python.org

If you don't have `pip` installed, you can:

- Download the script, from https://bootstrap.pypa.io/get-pip.py.
- Open a terminal/command prompt, `cd` to the folder containing the `get-pip.py` file and run:

Listing 3: Install pip in Windows environment.

```
> py get-pip.py
```

Listing 4: Install pip in macOS/Linux environment.

```
$ python3 get-pip.py
```

See also:

Read more details about this script in pypa/get-pip.

Read more about installation of `pip` in pip installation.

Install From PyPi Repository

`pip` is the recommended installer for ANLT Utility. The most common usage of `pip` is to install from the Python Package Index using Requirement Specifiers.

Note: If you install ANLT Utility using `pip install xoa-utils`, XOA Python API (PyPI package name xoa_driver) will be automatically installed.

Important: You can **either** install xoa-utils in your global Python namespace/package inventory **or** you can create a virtual Python environment to prevent polluting your global Python package inventory

If Install To Global Namespace

Listing 5: Install ANLT Utility in Windows environment from PyPI.

```
> pip install xoa-utils            # latest version
> pip install xoa-utils==1.0.0     # specific version
> pip install xoa-utils>=1.0.0     # minimum version
```

Listing 6: Install ANLT Utility in macOS/Linux environment from PyPI.

```
$ pip install xoa-utils            # latest version
$ pip install xoa-utils==1.0.0     # specific version
$ pip install xoa-utils>=1.0.0     # minimum version
```

If Install To Virtual Environment

Install ANLT Utility in a virtual environment, so it does not pollute your global namespace.

For example, your project folder is called `/my_xoa_project`.

Listing 7: Install ANLT Utility in a virtual environment in Windows from PyPI.

```
[my_xoa_project]> python -m venv .\env
[my_xoa_project]> .env\Scripts\activate

(env) [my_xoa_project]> pip install xoa-utils         # latest version
(env) [my_xoa_project]> pip install xoa-utils==1.0.0  # specific version
(env) [my_xoa_project]> pip install xoa-utils>=1.0.0  # minimum version
```

Listing 8: Install ANLT Utility in a virtual environment in macOS/Linux from PyPI.

```
[my_xoa_project]$ python3 -m venv ./env
[my_xoa_project]$ source ./env/bin/activate

(env) [my_xoa_project]$ pip install xoa-utils         # latest version
(env) [my_xoa_project]$ pip install xoa-utils==1.0.0  # specific version
(env) [my_xoa_project]$ pip install xoa-utile>=1.0.0 # minimum version
```

Afterwards, your project folder will be:

Listing 9: After creating Python virtual environment

```
/my_xoa_project
    |
    |- env
```

See also:

- Virtual Python environment
- virtualenv
- venv

Upgrade From PyPi Repository

To upgrade ANLT Utility package from PyPI:

Listing 10: Upgrade ANLT Utility in Windows environment from PyPI.

```
> pip install xoa-utils --upgrade
```

Listing 11: Upgrade ANLT Utility in macOS/Linux environment from PyPI.

```
$ pip install xoa-utils --upgrade
```

Note: If you upgrade ANLT Utility using `pip install --upgrade xoa-utils`, XOA Python API (PyPI package name xoa_driver) will be automatically upgraded.

Install Manually From Source

If you don't have internet access, you can install ANLT Utility manually from source, the steps are:

Step 1, make sure Python packages wheel and setuptools are installed on your system. Install `wheel` and `setuptools` using `pip`:

Listing 12: Install `wheel` and `setuptools` in Windows environment.

```
> pip install wheel setuptools
```

Listing 13: Install `wheel` and `setuptools` in macOS/Linux environment.

```
$ pip install wheel setuptools
```

Step 2, download the ANLT Utility source distribution from ANLT Utility Releases. Unzip the archive and run the `setup.py` script to install the package:

Listing 14: Install ANLT Utility in Windows environment from source.

```
[xoa_core]> python setup.py install
```

Listing 15: Install ANLT Utility in macOS/Linux environment from source.

```
[xoa_core]$ python3 setup.py install
```

Step 3, if you want to distribute, you can build `.whl` file for distribution from the source:

Listing 16: Build ANLT Utility wheel in Windows environment for distribution.

```
[xoa_core]> python setup.py bdist_wheel
```

Listing 17: Build ANLT Utility wheel in macOS/Linux environment for distribution.

```
[xoa_core]$ python3 setup.py bdist_wheel
```

Important: If you install ANLT Utility from the source code, you need to install XOA Python API (PyPI package name xoa_driver) separately. This is because XOA Python API is treated as a 3rd-party dependency of ANLT Utility. You can go to XOA Python API repository to learn how to install it.

Start ANLT Utility

After installing the package and ensuring the SSH key in place, you can start ANLT Utility simply by typing `xoa-utils`

Listing 18: Start ANLT Utility SSH service.

```
> xoa-utils
(PID: 12345) ANLT Utility SSH Service (1.1.0) running on 0.0.0.0:22622.
```

Listing 19: Start ANLT Utility SSH service.

```
$ xoa-utils
(PID: 12345) ANLT Utility SSH Service (1.1.0) running on 0.0.0.0:22622.
```

Note: If you want to run xoa-utils SSH service on a different port, do `xoa-utils <port number>`

Uninstall and Remove Unused Dependencies

`pip uninstall xoa-utils` can uninstall the package itself but not its dependencies. Leaving the package's dependencies in your environment can later create conflicting dependencies problem.

We recommend install and use the pip-autoremove utility to remove a package plus unused dependencies.

Listing 20: Uninstall ANLT Utility in Windows environment.

```
> pip install pip-autoremove
> pip-autoremove xoa-utils -y
```

Listing 21: Uninstall ANLT Utility in macOS/Linux environment.

```
$ pip install pip-autoremove
$ pip-autoremove xoa-utils -y
```

See also:

See the pip uninstall reference.

See pip-autoremove usage.

2.2 Step-by-Step Guide

This section provides a step-by-step guide on how to use ANLT Utility to do interactive ANLT test.

Note: You can use **tab key** to auto-complete a command to speed up your input speed.

2.2.1 SSH to ANLT Utility

After running the ANLT Utility SSH Server, use another console to SSH to ANLT Utility.

Listing 22: SSH to ANLT Utility.

```
> ssh yourname@localhost -p 22622

Hello yourname, welcome to Xena OpenAutomation ANLT Utility server (2.2.0)

xoa-utils >
```

Listing 23: SSH to ANLT Utility.

```
$ ssh yourname@localhost -p 22622

Hello yourname, welcome to Xena OpenAutomation ANLT Utility server (2.2.0)

xoa-utils >
```

2.2.2 Connect

First, you need to connect to your tester using the command *connect*.

If you don't know which ports you will use at the time of connecting to the port, just leave the option `--ports` empty as the example shows below. You can reserve ports later.

```
xoa-utils > connect 10.10.10.10 yourname
```

2.2.3 Reserve Port

Then, reserve a port on the tester using the command *port*, as shown in the example below.

Note: You can only work on one port at a time in one console window. If you want to simultaneously work on multiple ports, you can open multiple console windows.

```
xoa-utils[123456] > port 0/0
```

2.2.4 Start ANLT Logging

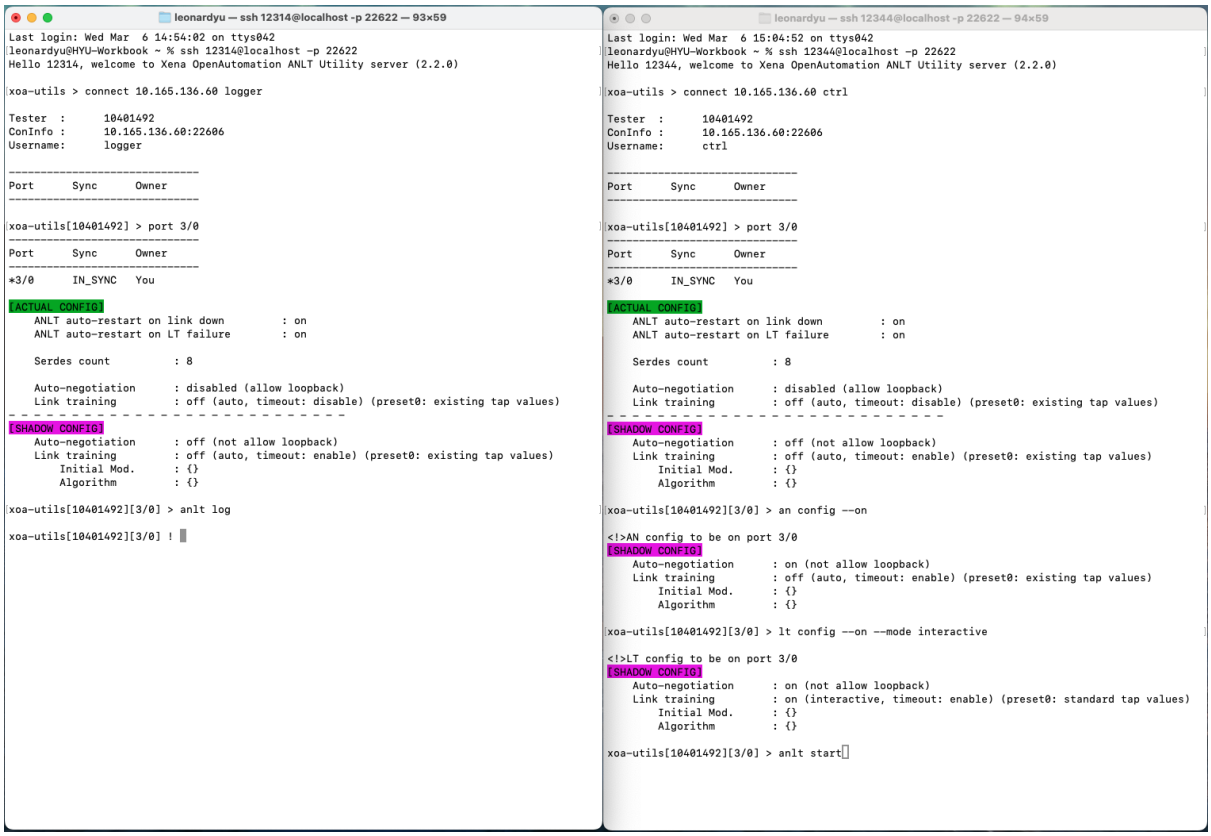
Start ANLT logging by *anlt log*.

```
xoa-utils[123456][0/0] > anlt log -f mylog.log
```

Note: This commands **continuously displays** the log messages on the screen so you can keep track of your ANLT actions.

To **quit** the continuous display mode, press `Control-z`.

Use one terminal to view the ANLT protocol trace, and use another to execute ANLT commands, as shown in the screenshot below.



2.2.5 Set ANLT Shadow Configuration

After disabling link recovery on the port, you can start setting ANLT shadow configuration using *an config*, *lt config*, *lt im*, and *lt alg* as the example shown below.

```
xoa-utils[123456][0/0] > an config --off --no-loopback

xoa-utils[123456][0/0] > lt config --on --preset0 ieee --mode interactive
```

Note: The initial modulation of each serdes on a port is by default PAM2 (NRZ). If you want to change them, you can use *lt im*, otherwise do nothing.

Important: *an config*, *lt config*, and *lt im* **only change the shadow ANLT configuration**. **To apply the configuration**, you need to run *anlt start*, otherwise your changes will not take effect on the tester.

2.2.6 Start ANLT

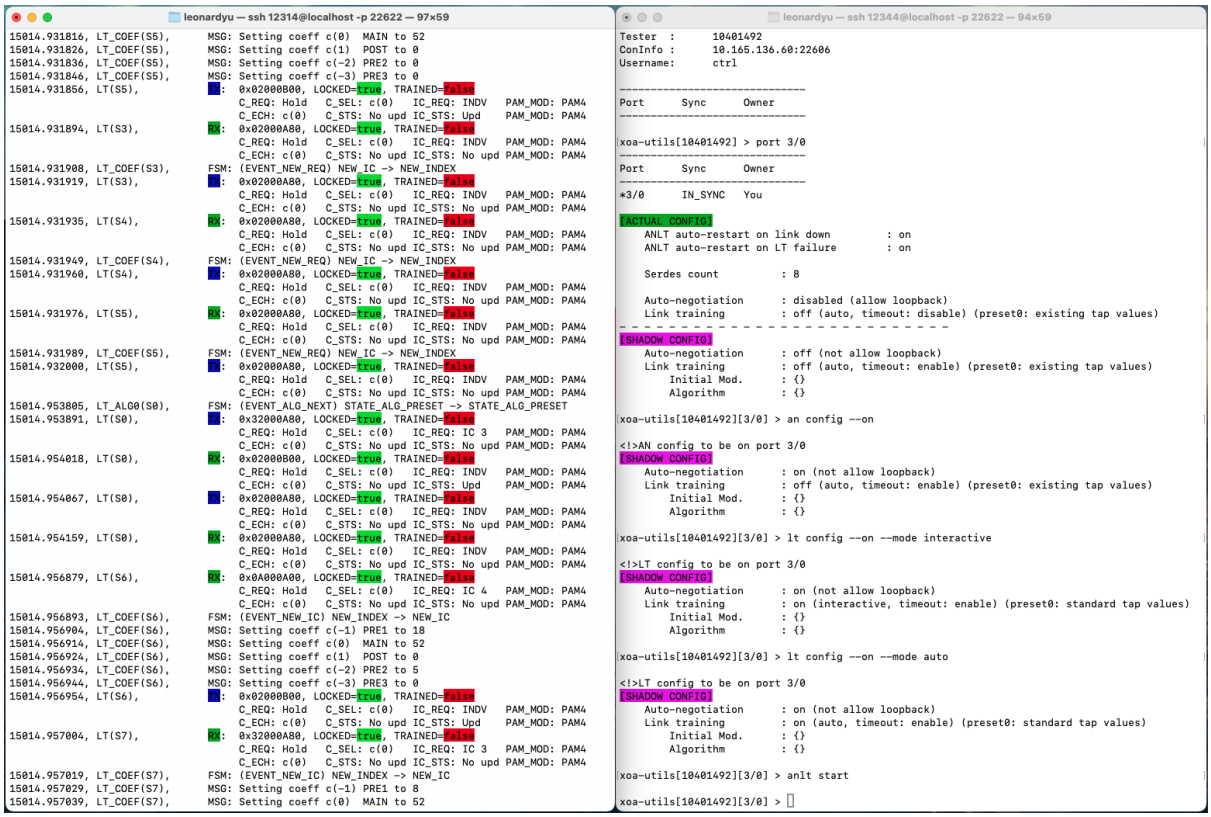
After configuring the ANLT shadow configuration, you should execute *anlt start* to **apply the shadow configuration and let the ANLT tester to start the ANLT procedure**.

See also:

Auto-Negotiation and Link Training Sequence

```
xoa-utils[123456][0/0] > anlt start
```

Use one terminal to view the ANLT protocol trace, and use another to execute ANLT commands, as shown in the sreenshot below.



2.2.7 Control LT Interactive

If you run LT (interactive), you will need to manually control the LT parameters using the LT Control Commands shown in *LT Group*, for example:

```
xoa-utils[123456][0/0] > lt preset 0 2

xoa-utils[123456][0/0] > lt inc 0 pre3

xoa-utils[123456][0/0] > lt inc 0 main

xoa-utils[123456][0/0] > lt dec 0 post

xoa-utils[123456][0/0] > lt status 0

xoa-utils[123456][0/0] > lt trained 0

xoa-utils[123456][0/0] > lt txtapget 0

xoa-utils[123456][0/0] > lt txtapset 0 0 0 1 56 0
```

2.2.8 Check AN Status

Check AN statistics by *an status*.

```
xoa-utils[123456][0/0] > an status

[AN STATUS]
    Mode                  : enabled
    Loopback              : allowed

    Duration              : 2,068,747 µs
    Successful runs       : 1
    Timeouts              : 0
    Loss of sync          : 0

    HCD                   : IEEE_800GBASE_CR8_KR8
    HCD negotiation fails : 0
    FEC result            : RS_FEC_KP
    FEC negotiation fails : 0

                                RX    TX
    Link codewords        :      2     1
    Next-page messages    :      0     0
    Unformatted pages     :      0     0
```

2.2.9 Check LT Status

Check LT statistics by *lt status*.

```
xoa-utils[123456][0/0] > lt status 0

[LT STATUS]
    Is enabled        : true
    Is trained        : true
    Failure           : no_failure

    Initial mod.      : nrz
    Preset0           : standard tap values
    Total bits        : 2,201,372,480
    Total err. bits   : 24
    BER               : 1.09e-08

    Duration          : 2,000,250 µs

    Lock lost         : 2
    Frame lock        : locked
    Remote frame lock : locked

    Frame errors      : 1
    Overrun errors    : 1

    Last IC received  : Preset 3
    Last IC sent      : Preset 3

    TX Coefficient              :          c(-3)       c(-2)       c(-1)        c(0) ␣
↪       c(1)
        Current level           :             0           0           1          44 ␣
↪          0
                                :         RX  TX      RX  TX      RX  TX      RX  TX ␣
↪     RX  TX
          + req                 :          0   0       0   0       2   2       1   1 ␣
↪      0   0
          - req                 :          0   0       0   0       2   2       0   0 ␣
↪      0   0
          coeff/eq limit reached  :          0   0       0   0       0   0       0   0 ␣
↪      0   0
          eq limit reached      :          0   0       0   0       0   0       0   0 ␣
↪      0   0
          coeff not supported   :          0   0       0   0       0   0       0   0 ␣
↪      0   0
          coeff at limit        :          0   0       0   0       0   0       0   0 ␣
↪      0   0
```

2.2.10 Stop ANLT and Restart

To stop and start ANLT again:

```
xoa-utils[123456][0/0] > anlt stop

xoa-utils[123456][0/0] > anlt start
```

CHAPTER

THREE

INTERPRETING ANLT LOG

3.1 Save or Read Log

With Freya, you not only have the capability to manually control the link training process but also gain visibility into the ANLT protocol by reading the protocol trace log file.

- Use `anlt log -f <filename>.log` to display and save the **live log** from Freya port during ANLT.

Note: You need first execute *port* to switch your working port before collecting log on it.

- Use `anlt log --read -f <saved_filename>.log` to read the **already saved** log file.

You can use ANLT Utility as a log file reader. Reading a save log file doesn't need to reserve a port or connect to the chassis, as shown in the screenshot below.

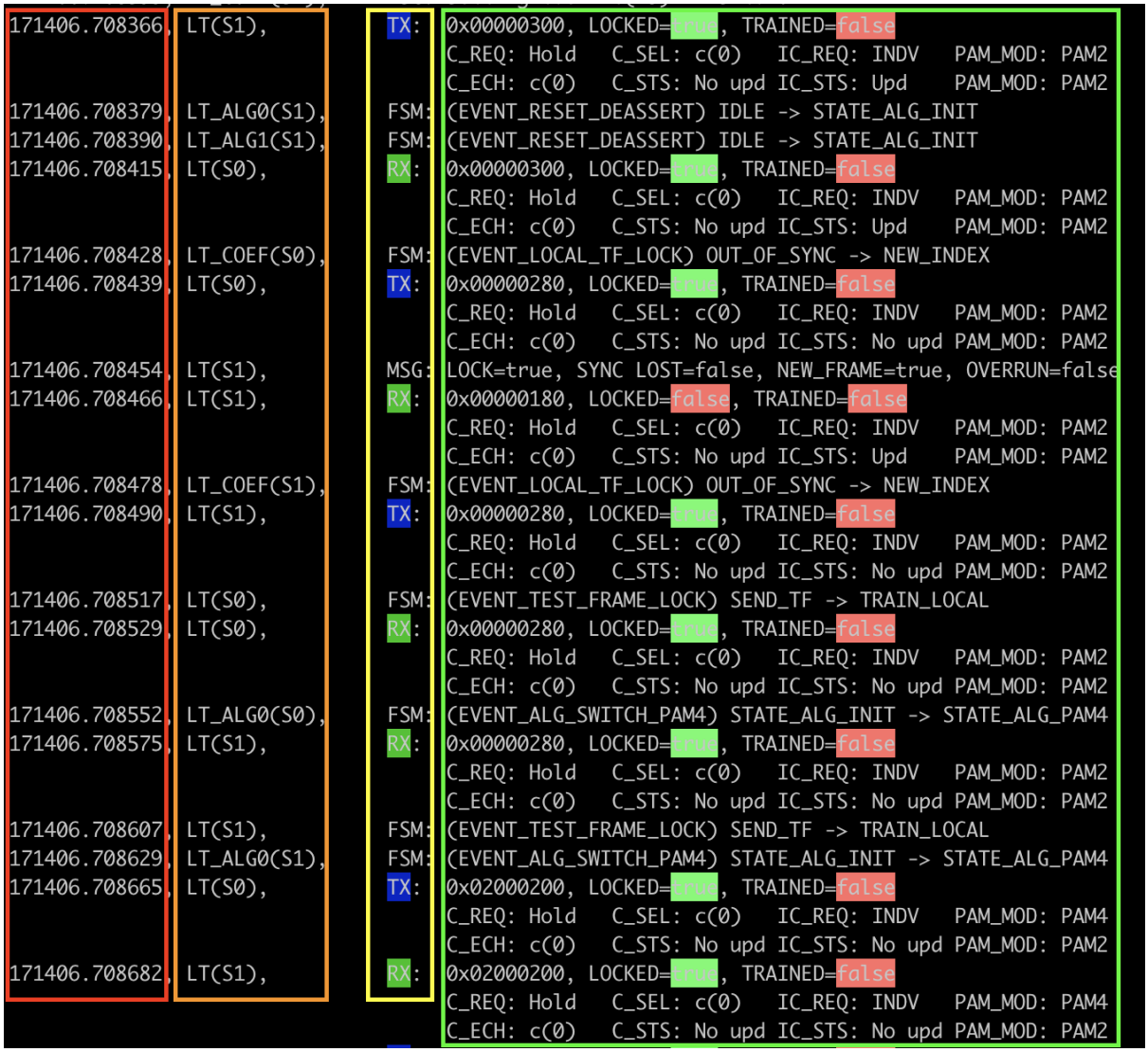
```
Last login: Sun Mar 17 12:35:57 on ttys014
leonardyu@HRW-L000058 ~ % ssh 2342@localhost -p 22622
Hello 2342, welcome to Xena OpenAutomation ANLT Utility server (2.2.0)

xoa-utils > anlt log --read -f 20240317_1.log
8.812708, LT(S0),               FSM: (EVENT_RESET_DEASSERT) IDLE -> INITIALIZE
8.814008, LT(S1),               FSM: (EVENT_RESET_DEASSERT) IDLE -> INITIALIZE
8.814119, LT(S0),               FSM: (EVENT_RESET_DEASSERT) IDLE -> INITIALIZE
8.814128, LT(S1),               FSM: (EVENT_RESET_DEASSERT) IDLE -> INITIALIZE
8.814165, ANEG,                 FSM: (EVENT_INIT_DONE) IDLE -> WAIT_ANEG_ENABLE
```

Note: You should **place the log file in the same directory** where the ANLT Utility `.exe` is.

3.2 Structure

A log message consists of 4 parts

1. Timestamp (red block)
2. Protocol (orange block)
3. Message Type (yellow block)
4. Message Content (green block)


171406.708366, LT(S1), TX: 0x00000300, LOCKED=true, TRAINED=false
C_REQ: Hold C_SEL: c(0) IC_REQ: INDV PAM_MOD: PAM2
C_ECH: c(0) C_STS: No upd IC_STS: Upd PAM_MOD: PAM2
171406.708379, LT_ALG0(S1), FSM: (EVENT_RESET_DEASSERT) IDLE -> STATE_ALG_INIT
171406.708390, LT_ALG1(S1), FSM: (EVENT_RESET_DEASSERT) IDLE -> STATE_ALG_INIT
171406.708415, LT(S0), RX: 0x00000300, LOCKED=true, TRAINED=false
C_REQ: Hold C_SEL: c(0) IC_REQ: INDV PAM_MOD: PAM2
C_ECH: c(0) C_STS: No upd IC_STS: Upd PAM_MOD: PAM2
171406.708428, LT_COEF(S0), FSM: (EVENT_LOCAL_TF_LOCK) OUT_OF_SYNC -> NEW_INDEX
171406.708439, LT(S0), TX: 0x00000280, LOCKED=true, TRAINED=false
C_REQ: Hold C_SEL: c(0) IC_REQ: INDV PAM_MOD: PAM2
C_ECH: c(0) C_STS: No upd IC_STS: No upd PAM_MOD: PAM2
171406.708454, LT(S1), MSG: LOCK=true, SYNC LOST=false, NEW_FRAME=true, OVERRUN=false
171406.708466, LT(S1), RX: 0x00000180, LOCKED=false, TRAINED=false
C_REQ: Hold C_SEL: c(0) IC_REQ: INDV PAM_MOD: PAM2
C_ECH: c(0) C_STS: No upd IC_STS: Upd PAM_MOD: PAM2
171406.708478, LT_COEF(S1), FSM: (EVENT_LOCAL_TF_LOCK) OUT_OF_SYNC -> NEW_INDEX
171406.708490, LT(S1), TX: 0x00000280, LOCKED=true, TRAINED=false
C_REQ: Hold C_SEL: c(0) IC_REQ: INDV PAM_MOD: PAM2
C_ECH: c(0) C_STS: No upd IC_STS: No upd PAM_MOD: PAM2
171406.708517, LT(S0), FSM: (EVENT_TEST_FRAME_LOCK) SEND_TF -> TRAIN_LOCAL
171406.708529, LT(S0), RX: 0x00000280, LOCKED=true, TRAINED=false
C_REQ: Hold C_SEL: c(0) IC_REQ: INDV PAM_MOD: PAM2
C_ECH: c(0) C_STS: No upd IC_STS: No upd PAM_MOD: PAM2
171406.708552, LT_ALG0(S0), FSM: (EVENT_ALG_SWITCH_PAM4) STATE_ALG_INIT -> STATE_ALG_PAM4
171406.708575, LT(S1), RX: 0x00000280, LOCKED=true, TRAINED=false
C_REQ: Hold C_SEL: c(0) IC_REQ: INDV PAM_MOD: PAM2
C_ECH: c(0) C_STS: No upd IC_STS: No upd PAM_MOD: PAM2
171406.708607, LT(S1), FSM: (EVENT_TEST_FRAME_LOCK) SEND_TF -> TRAIN_LOCAL
171406.708629, LT_ALG0(S1), FSM: (EVENT_ALG_SWITCH_PAM4) STATE_ALG_INIT -> STATE_ALG_PAM4
171406.708665, LT(S0), TX: 0x02000200, LOCKED=true, TRAINED=false
C_REQ: Hold C_SEL: c(0) IC_REQ: INDV PAM_MOD: PAM4
C_ECH: c(0) C_STS: No upd IC_STS: No upd PAM_MOD: PAM2
171406.708682, LT(S1), RX: 0x02000200, LOCKED=true, TRAINED=false
C_REQ: Hold C_SEL: c(0) IC_REQ: INDV PAM_MOD: PAM4
C_ECH: c(0) C_STS: No upd IC_STS: No upd PAM_MOD: PAM2

3.2.1 Timestamp

The timestamp unit is in seconds, for example, `171406.514179` represents 171406.514179 seconds. The granularity is in microseconds.

When analyzing your ANLT log trace, focus on the relative time difference between messages rather than solely concentrating on the absolute time value. The absolute time value describes the elapsed time since the chassis was started.

3.2.2 Protocol

`ANEG`: message is generated by auto-negotiation.

`LT(S<x>)`: message is generated by link training serdes lane <x>.

`LT_COEFF(S<x>)`: message is generated by link training coefficient on serdes lane <x>.

3.2.3 Message Type

`FSM`: Finite State Machine type.

`MSG`: log message type.

`TX`: ANLT test frame transmitted by the port to the remote port.

`RX`: ANLT test frame from the remote port received by the port.

3.2.4 Message Content

For messages of type `FSM`, the message content includes the state information or state transition information of the Finite State Machine.

For messages of type `MSG`, the message content shows the log message.

For messages of type `TX` and `RX`, the message content includes both the raw Hex value of the test frame and its human-readable meaning.

3.3 Explanation

3.3.1 ANEG FSM

```
171406.515335, ANEG,            FSM: (EVENT_INIT_DONE) IDLE -> WAIT_ANEG_ENABLE
171406.515382, ANEG,            FSM: (EVENT_AUTONEG_DISABLE) WAIT_ANEG_ENABLE -> AN_
↪GOOD_CHECK
```

ANEG FSM messages show the FSM state and state transition of ANEG defined in IEEE 802.3-2022 73.10.4, as shown in the screenshots below.

73.10.4 State diagrams

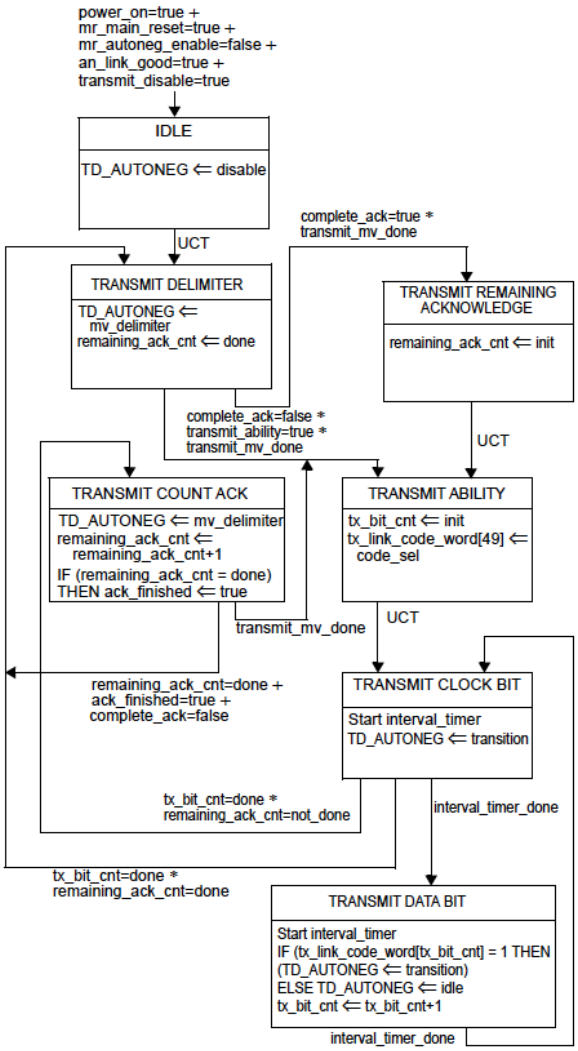


Figure 73–9—Transmit state diagram

IEEE Std 802.3-2022, IEEE Standard for Ethernet
SECTION FIVE

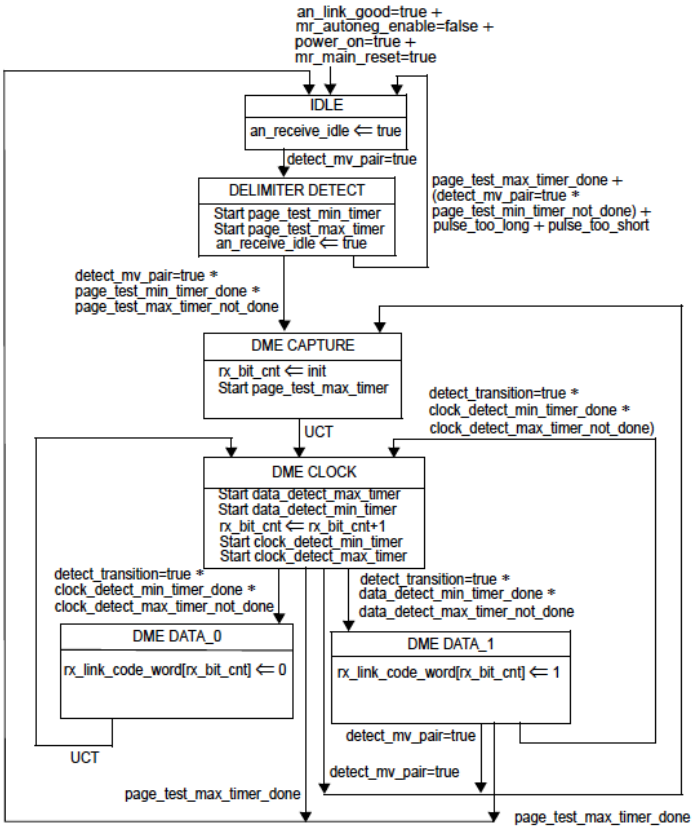


Figure 73–10—Receive state diagram

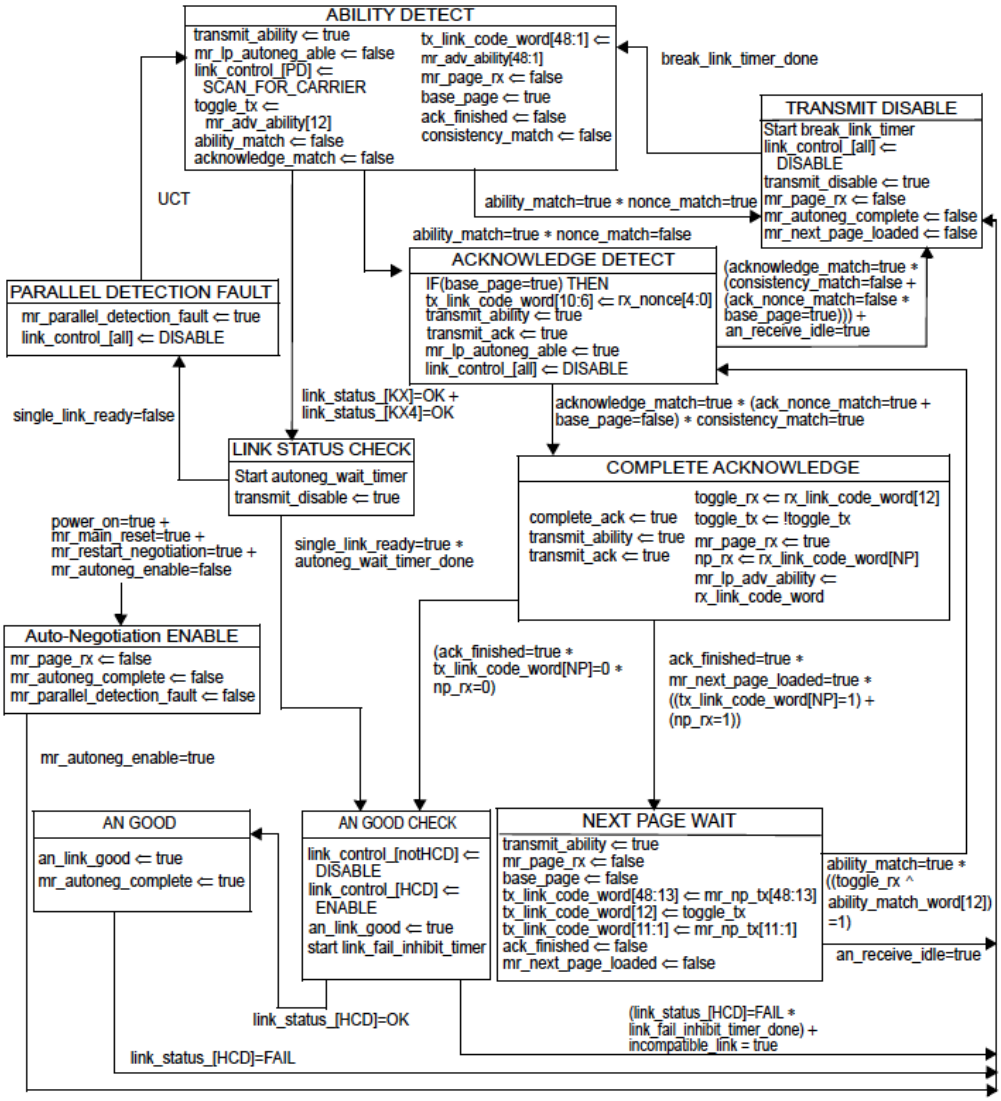


Figure 73–11—Arbitration state diagram

3.3.2 ANEG MSG

```
171406.654505, ANEG,            MSG: SYNC=false, SYNC LOST=true, NEW_PAGE=true
```

ANEG MSG messages show log messages from ANEG. Explanations of the message content:

- SYNC: indicates the current lock status. SYNC=true means locked. SYNC=false means lock lost.
- SYNC LOST: indicates the previous lock status. SYNC LOST=true means the port lost lock previously. SYNC LOST=false means the port locked previously.
- NEW_PAGE: indicates if there is New Page.

3.3.3 ANEG TX & RX

The raw hex value of the transmitted and received ANEG test frames are shown first. Then base page and next page indicate if it is Base Page or Next Page, followed by the value of each field.

```
171406.519452, ANEG,            MSG: TRANSMIT_DISABLE - ANEG restart
171406.519462, ANEG,            TX:  0x004000198001, base page, NP:1, ACK:0, RF:0,␣
↪TN:25, EN:0, C:0
                                     FEC:[], ABILITY:['200GBASE_KR2_CR2']
171406.519481, ANEG,            RX:  0x004000198001, base page, NP:1, ACK:0, RF:0,␣
↪TN:25, EN:0, C:0
                                     FEC:[], ABILITY:['200GBASE_KR2_CR2']
171406.586888, ANEG,            FSM: (EVENT_BREAK_LINK_TIMER_DONE) TRANSMIT_DISABLE ->
↪ ABILITY_DETECT
171406.586905, ANEG,            MSG: SYNC=true, SYNC LOST=false, NEW_PAGE=true
171406.586917, ANEG,            TX:  0x004000198001, base page, NP:1, ACK:0, RF:0,␣
↪TN:25, EN:0, C:0
                                     FEC:[], ABILITY:['200GBASE_KR2_CR2']
171406.586935, ANEG,            RX:  0x004000198001, base page, NP:1, ACK:0, RF:0,␣
↪TN:25, EN:0, C:0
                                     FEC:[], ABILITY:['200GBASE_KR2_CR2']
171406.586984, ANEG,            FSM: (EVENT_ABILITY_MATCH_NONCE) ABILITY_DETECT ->␣
↪TRANSMIT_DISABLE
```

```
171406.654806, ANEG,            FSM: (EVENT_ACK_NP) COMPLETE_ACKNOWLEDGE -> NEXT_PAGE_
↪WAIT
171406.654818, ANEG,            TX:  0x04DF0353A805, next page, NP:1, ACK:0, MP:1,␣
↪ACK2:0, T:1
                                     Formatted message:
                                     Value:0x0005, Msg:OUI Tagged: 0x6a737c␣
↪(preliminary)
171406.654837, ANEG,            RX:  0x04DF0353A805, next page, NP:1, ACK:0, MP:1,␣
↪ACK2:0, T:1
                                     Formatted message:
                                     Value:0x0005, Msg:OUI Tagged: 0x6a737c␣
↪(preliminary)
171406.654889, ANEG,            FSM: (EVENT_NEXT_PAGE) NEXT_PAGE_WAIT -> ACKNOWLEDGE_
↪DETECT
171406.654901, ANEG,            TX:  0x04DF0353E805, next page, NP:1, ACK:1, MP:1,␣
↪ACK2:0, T:1
                                     Formatted message:
                                     Value:0x0005, Msg:OUI Tagged: 0x6a737c␣
↪(preliminary)
171406.654939, ANEG,            RX:  0x04DF0353E805, next page, NP:1, ACK:1, MP:1,␣
↪ACK2:0, T:1
```

(continues on next page)

(continued from previous page)

```
                                    Formatted message:
                                    Value:0x0005, Msg:OUI Tagged: 0x6a737c␣
↪(preliminary)
```

Base Page

Base Page is defined in IEEE 802.3-2022 73.6

73.6 Link codeword encoding

The base link codeword (Base Page) transmitted within a DME page shall convey the encoding shown in Figure 73–6. The Auto-Negotiation function supports additional pages using the Next Page function. Encoding for the link codeword(s) used in the Next Page exchange are defined in 73.7.7. In a DME page, D0 shall be the first bit transmitted.

3098
Copyright © 2022 IEEE. All rights reserved.
Authorized licensed use limited to: Claus Hoyer. Downloaded on August 23,2023 at 08:40:58 UTC from IEEE Xplore. Restrictions apply.

D 0	D 1	D 2	D 3	D 4	D 5	D 6	D 7	D 8	D 9	D 10	D 11	D 12	D 13	D 14	D 15
S 0	S 1	S 2	S 3	S 4	E 0	E 1	E 2	E 3	E 4	C 0	C 1	C 2	RF	Ack	NP

D 16	D 17	D 18	D 19	D 20	D 21	D 22	D 23	D 24	D 25	D 26	D 43	D 44	D 45	D 46	D 47
T 0	T 1	T 2	T 3	T 4	A 0	A 1	A 2	A 3	A 4	A 21	A 22	F 2	F 3	F 0	F 1

Figure 73–6—Link codeword Base Page

D[4:0] contains the Selector Field. D[9:5] contains the Echoed Nonce field. D[12:10] contains capability bits to advertise capabilities not related to the PHY. C[1:0] is used to advertise pause capability. The remaining capability bit C[2] is reserved. D[15:13] contains the RF, Ack, and NP bits. These bits shall function as specified in 28.2.1.2. D[20:16] contains the Transmitted Nonce field. D[43:21] contains the Technology Ability Field. D[47:44] contains FEC capability (see 73.6.5).

- `NP`: Next Page, IEEE 802.3-2022 73.6.9

73.6.9 Next Page

Next Page (NP) is encoded in bit D15 of link codeword. Support of Next Pages is mandatory. If the device does not have any Next Pages to send, the NP bit shall be set to logical zero. If a device wishes to engage in Next Page exchange, it shall set the NP bit to logical one. If a device has no Next Pages to send and its link partner has set the NP bit to logical one, it shall transmit Next Pages with Null message codes and the NP bit set to logical zero while its link partner transmits valid Next Pages. Next page exchanges will occur if either the device or its link partner sets the Next Page bit to logical one. The Next Page function is defined in 73.7.7.

- `ACK`: Acknowledge, IEEE 802.3-2022 73.6.8
- `RF`: Remote Fault, IEEE 802.3-2022 73.6.7

73.6.8 Acknowledge

Acknowledge (Ack) is used by the Auto-Negotiation function to indicate that a device has successfully received its link partner's link codeword. The Acknowledge Bit is encoded in bit D14 of link codeword. If no Next Page information is to be sent, this bit shall be set to logical one in the link codeword after the reception of at least three consecutive and consistent DME pages (ignoring the Acknowledge bit value). If Next Page information is to be sent, this bit shall be set to logical one after the device has successfully received at least three consecutive and matching DME pages (ignoring the Acknowledge bit value), and will remain set until the Next Page information has been loaded into the AN XNP transmit register (registers 7.22, 7.23, 7.24). In order to save the current received link codeword, it has to be read from the AN LP XNP ability register (register 7.25, 7.26, 7.27) before the Next Page of transmit information is loaded into the AN XNP transmit register. After the COMPLETE ACKNOWLEDGE state has been entered, the link codeword will be transmitted at least six times.

73.6.7 Remote Fault

Remote Fault (RF) is encoded in bit D13 of the base link codeword. The default value is logical zero. The Remote Fault bit provides a standard transport mechanism for the transmission of simple fault information. When the RF bit in the AN advertisement register (register 7.16.13) is set to logical one, the RF bit in the transmitted base link codeword is set to logical one. When the RF bit in the received base link codeword is set to logical one, the Remote Fault bit in the AN LP Base Page ability register (register 7.19.13) will be set to logical one, if the management function is present.

- `TN`: Transmitted Nonce Field, IEEE 802.3-2022 73.6.3

73.6.3 Transmitted Nonce Field

Transmitted Nonce Field (T[4:0]) is a 5-bit wide field containing a random or pseudo-random number. A new value shall be generated for each entry to the Ability Detect state. The method of generating the nonce is left to the implementer. The transmitted nonce should have a uniform distribution in the range from 0 to $2^5 - 1$. The method used to generate the value should be designed to minimize correlation to the values generated by other devices.

- `EN`: Echoed Nonce, IEEE 802.3-2022 73.6.2
- `C`: Pause Ability, IEEE 802.3-2022 73.6.6
- `FEC`: FEC capability, IEEE 802.3-2022 73.6.5
- `ABILITY`: Technology Ability, IEEE 802.3-2022 73.6.4

Next Page

Next Page is defined in IEEE 802.3-2022 73.7.7. Next Page contains the following fields

- `ACK`: Acknowledge
- `ACK2`: Acknowledge 2
- `MP`: Message Page
- `T`: Toggle

73.6.2 Echoed Nonce Field

Echoed Nonce Field (E[4:0]) is a 5-bit wide field containing the nonce received from the link partner. When Acknowledge is set to logical zero, the bits in this field shall contain logical zeros. When Acknowledge is set to logical one, the bits in this field shall contain the value received in the Transmitted Nonce Field from the link partner.

73.6.6 Pause Ability

Pause (C0:C1) is encoded in bits D11:D10 of the base link codeword. The two-bit Pause is encoded as follows:

a) C0 is the same as PAUSE as defined in Annex 28B
b) C1 is the same as ASM_DIR as defined in Annex 28B

Copyright © 2022 IEEE. All rights reserved.
Authorized licensed use limited to: Claus Hoyer. Downloaded on August 23,2023 at 08:40:58 UTC from IEEE Xplore. Restrictions apply.

The Pause encoding is defined in Clause 28B.2, Table 28B–2. The PAUSE bit indicates that the device is capable of providing the symmetric PAUSE functions as defined in Annex 31B. The ASM_DIR bit indicates that asymmetric PAUSE is supported. The value of the PAUSE bit when the ASM_DIR bit is set indicates the direction the PAUSE frames are supported for flow across the link. Asymmetric PAUSE configuration results in independent enabling of the PAUSE receive and PAUSE transmit functions as defined by Annex 31B. See 28B.3 regarding PAUSE configuration resolution.

73.6.5 FEC capability

FEC (F2, F3, F0, F1) is encoded in bits D44:D47 of the base link codeword. The four FEC bits are used as follows:

3100
Copyright © 2022 IEEE. All rights reserved.
Authorized licensed use limited to: Claus Hoyer. Downloaded on August 23,2023 at 08:40:58 UTC from IEEE Xplore. Restrictions apply.

a) F0 is 10 Gb/s per lane FEC ability
b) F1 is 10 Gb/s per lane FEC requested
c) F2 is 25G RS-FEC requested
d) F3 is 25G BASE-R FEC requested

Bits F2 and F3 are used for resolving FEC operation for 25G PHYs, while bits F0 and F1 are used for 10 Gb/s per lane operation. Bits F0 and F1 are not used for 25G PHYs.

73.6.5.1 FEC resolution for 25G PHYs

For 25G PHYs if neither PHY requests FEC operation in bits F2 or F3 then FEC is not enabled.

For 25GBASE-KR and 25GBASE-CR PHYs if either PHY requests RS-FEC then RS-FEC operation is enabled, otherwise if either PHY requests BASE-R FEC then BASE-R operation is enabled.

For 25GBASE-KR-S and 25GBASE-CR-S PHYs, if either PHY requests RS-FEC or BASE-R FEC then BASE-R operation is enabled. This is because 25GBASE-KR-S and 25GBASE-CR-S PHYs do not support RS-FEC operation.

73.6.5.2 FEC resolution for 10 Gb/s per lane PHYs

For 10 Gb/s per lane operation, when the FEC ability bit F0 is set to logical one, it indicates that the PHY has FEC ability (see Clause 74). When the FEC requested F1 bit is set to logical one, it indicates a request to enable FEC on the link.

Since the local device and the link partner may have set the FEC capability bits differently, the priority resolution function is used to enable FEC in the respective PHYs. The FEC function shall be enabled on the link if 10GBASE-KR, 40GBASE-KR4, 40GBASE-CR4, or 100GBASE-CR10 is the HCD technology (see 73.7.6), both devices advertise FEC ability on the F0 bits, and at least one device requests FEC on the F1 bits; otherwise FEC shall not be enabled.

73.6.5.3 FEC control variables

The variable an_baser_fec_control indicates that BASE-R FEC operation has been negotiated. If the value is false, then BASE-R FEC has not been negotiated. If the value is true, then BASE-R FEC has been negotiated.

The variable an_rs_fec_control indicates that RS-FEC operation has been negotiated. If the value is false, then RS-FEC has not been negotiated. If the value is true, then RS-FEC has been negotiated.

The mapping of these variables to MDIO register bits is defined in Table 73–6.

If mr_autoneg_enable (see 73.10.1) is false, the FEC function is controlled by implementation-dependent means.

73.6.4 Technology Ability Field

Technology Ability Field (A[22:0]) is a 23-bit wide field containing information indicating supported technologies specific to the selector field value when used with the Auto-Negotiation for backplane and copper cable assembly. These bits are mapped to individual technologies such that abilities are advertised in parallel for a single selector field value. The Technology Ability Field encoding for the IEEE 802.3 selector with Auto-Negotiation for backplane and copper cable assembly is described in Table 73–4.

Table 73–4—Technology Ability Field encoding

Bit	Technology
A0	1000BASE-KX
A1	10GBASE-KX4
A2	10GBASE-KR
A3	40GBASE-KR4
A4	40GBASE-CR4
A5	100GBASE-CR10
A6	100GBASE-KP4
A7	100GBASE-KR4
A8	100GBASE-CR4
A9	25GBASE-KR-S or 25GBASE-CR-S
A10	25GBASE-KR or 25GBASE-CR
A11	2.5GBASE-KX
A12	5GBASE-KR
A13	50GBASE-KR or 50GBASE-CR
A14	100GBASE-KR2 or 100GBASE-CR2
A15	200GBASE-KR4 or 200GBASE-CR4
A16 through A22	Reserved

Multiple technologies may be advertised in the link codeword. A device shall support the data service ability for a technology it advertises. It is the responsibility of the Arbitration function to determine the common mode of operation shared by a link partner and to resolve multiple common modes.

NOTE—Previous editions of this standard prohibited simultaneous advertisement of PHYs that support operation over electrical backplanes with PHYs that support operation over copper cable assemblies.

25GBASE-KR-S abilities are a subset of 25GBASE-KR abilities, and likewise 25GBASE-CR-S abilities are a subset of 25GBASE-CR abilities. To allow interoperation between 25GBASE-KR-S and 25GBASE-KR PHY types, and between 25GBASE-CR-S and 25GBASE-CR PHY types, a device that supports 25GBASE-KR or 25GBASE-CR should advertise both A9 and A10 ability bits during auto-negotiation.

The fields A[22:16] are reserved for future use. Reserved fields shall be sent as zero and ignored on receive.

73.7.7 Next Page function

The Next Page function uses the Auto-Negotiation arbitration mechanisms to allow exchange of Next Pages of information, which may follow the transmission and acknowledgment procedures used for the base link codeword. The Next Page has both Message code field and Unformatted code fields.

Copyright © 2022 IEEE. All rights reserved.
Authorized licensed use limited to: Claus Hoyer. Downloaded on August 23,2023 at 08:40:58 UTC from IEEE Xplore. Restrictions apply.

A dual acknowledgment system is used. Acknowledge (Ack) is used to acknowledge receipt of the information; Acknowledge 2 (Ack2) is used to indicate that the receiver is able to act on the information (or perform the task) defined in the message.

The Toggle bit is used to ensure proper synchronization between the local device and the link partner.

Next page exchange occurs after the base link codewords have been exchanged if either end of the link segment set the Next Page bit to logical one indicating that it had at least one Next Page to send. Next page exchange consists of using the normal Auto-Negotiation arbitration process to send Next Page messages.

The Next Page contains two message encodings. The message encodings are defined as follows: message code, which contain predefined 11-bit codes, and unformatted code contains 32 bit codes. Multiple Next Pages with appropriate message codes and unformatted codes can be transmitted to send extended messages. Each series of Next Pages shall have a Message code that defines how the Unformatted codes will be interpreted. Any number of Next Pages may be sent in any order; however, it is recommended that the total number of Next Pages sent be kept small to minimize the link startup time.

Next page transmission ends when both ends of a link segment set their Next Page bits to logical zero, indicating that neither has anything additional to transmit. It is possible for one device to have more pages to transmit than the other device. Once a device has completed transmission of its Next Page information, it shall transmit Next Pages with Null message codes and the NP bit set to logical zero while its link partner continues to transmit valid Next Pages. An Auto-Negotiation able device shall recognize reception of Message Pages with Null message codes as the end of its link partner's Next Page information.

73.7.7.1 Next page encodings

The Next Page shall use the encoding shown in Figure 73–7 and Figure 73–8 for the NP, Ack, MP, Ack2, and T bits. These bits shall function as specified in 28.2.3.4. There are two types of Next Page encodings—message and unformatted. For message Next Pages, the MP bit shall be set to logical one, the 11-bit field D[10:0] shall be encoded as a Message Code Field and D[47:16] shall be encoded as Unformatted Code Field. For Unformatted Next Pages, the MP bit shall be set to logical zero; D[10:0] and D[47:16] shall be encoded as the Unformatted Code Field.

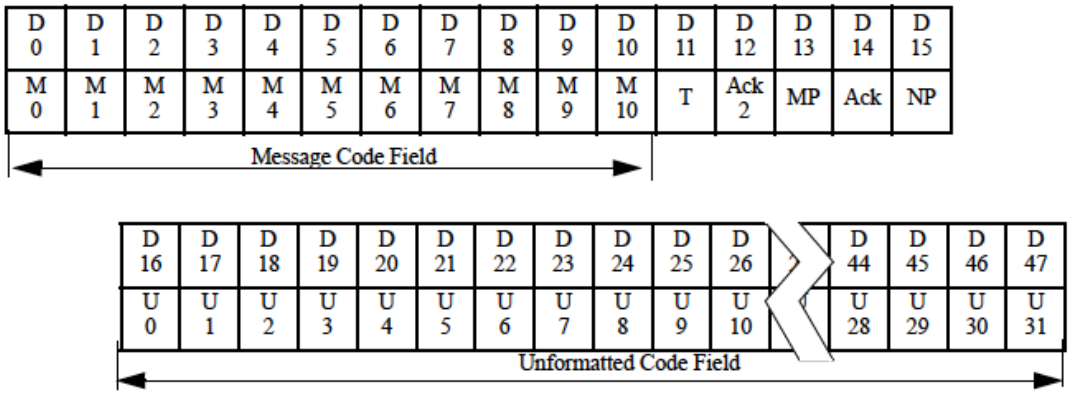


Figure 73–7—Message Next Page

Copyright © 2022 IEEE. All rights reserved.
Authorized licensed use limited to: Claus Hoyer. Downloaded on August 23,2023 at 08:40:58 UTC from IEEE Xplore. Restrictions apply.

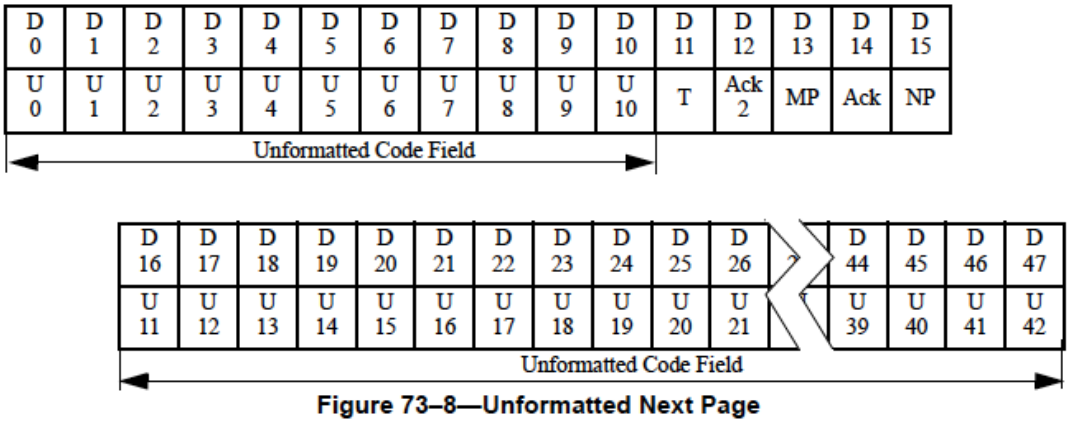


Figure 73–8—Unformatted Next Page

3.3.4 LT FSM

```
171406.515404, LT(S0),          FSM: (EVENT_TRAINING_ENABLE) INITIALIZE -> START_
↪DELAY1
171406.515426, LT(S1),          FSM: (EVENT_TRAINING_ENABLE) INITIALIZE -> START_
↪DELAY1
171406.518225, LT(S0),          FSM: (EVENT_RESET_DEASSERT) IDLE -> INITIALIZE
171406.519238, LT(S1),          FSM: (EVENT_RESET_DEASSERT) IDLE -> INITIALIZE
171406.519323, LT(S0),          FSM: (EVENT_RESET_DEASSERT) IDLE -> INITIALIZE
171406.519336, LT(S1),          FSM: (EVENT_RESET_DEASSERT) IDLE -> INITIALIZE
```

LT FSM messages show the FSM state transition of LT for each serdes lane, e.g. `LT(S0)` for lane 0, and `LT(S1)` for lane 1.

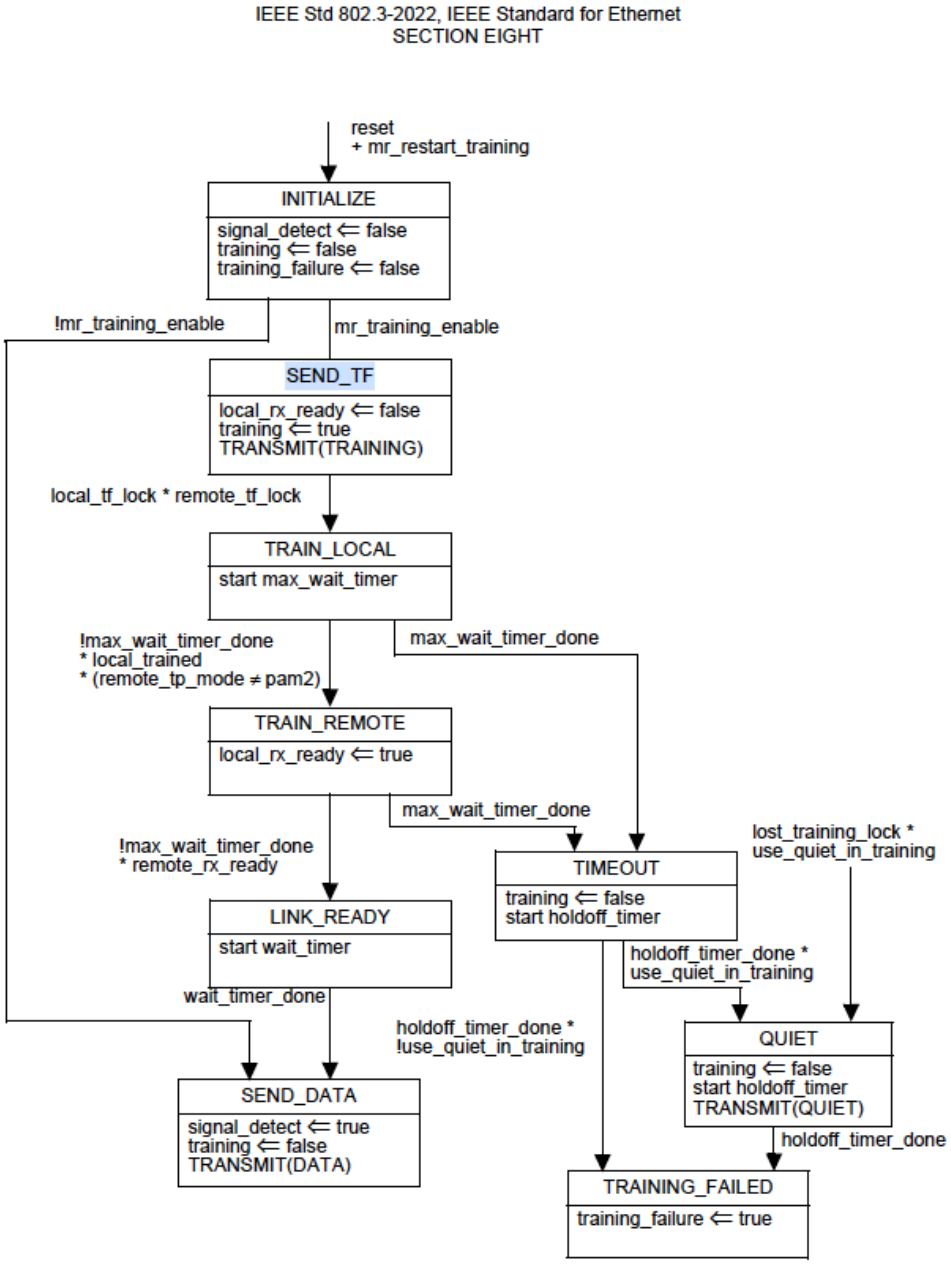


Figure 136–7—PMD control state diagram

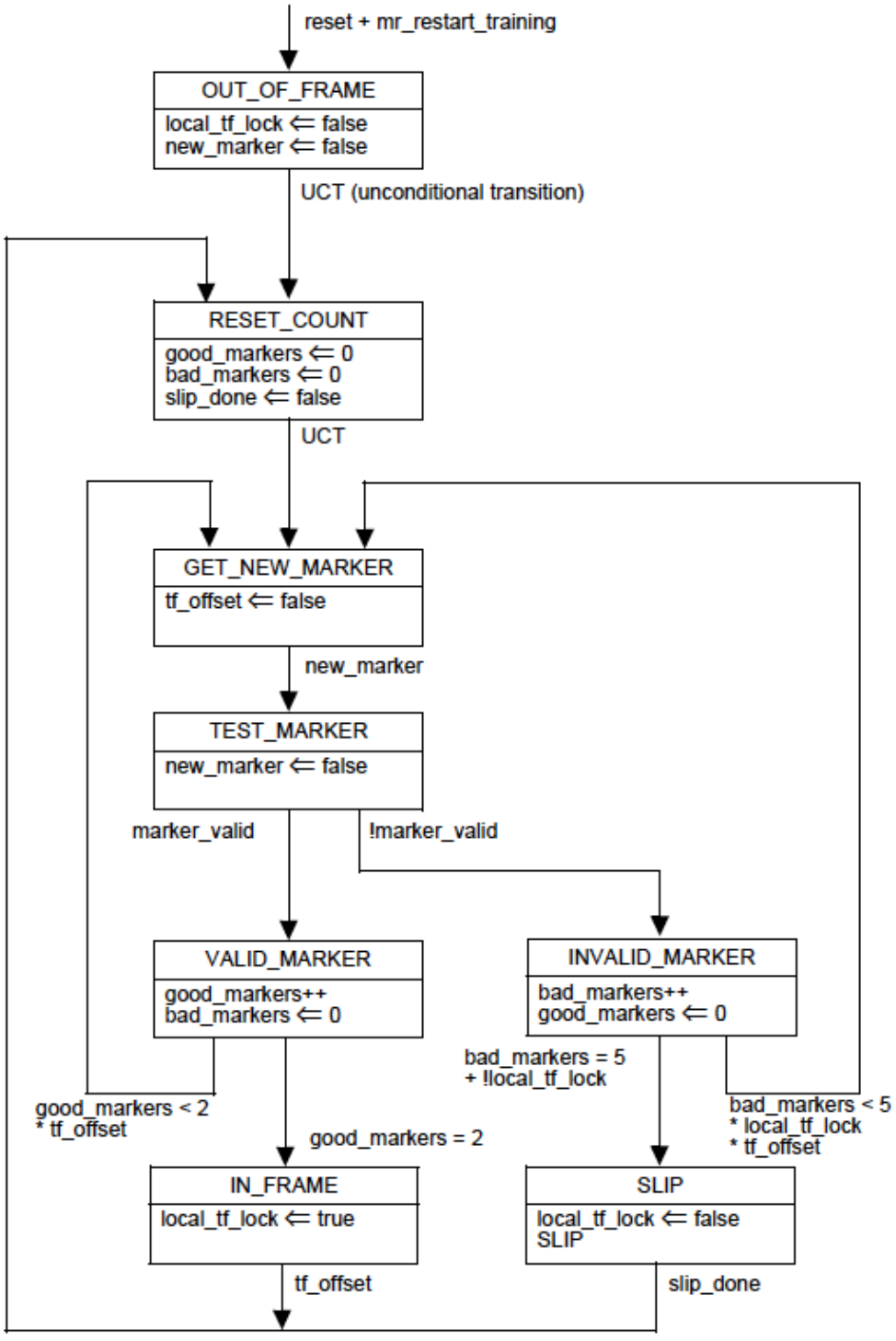


Figure 136–8—Training frame lock state diagram

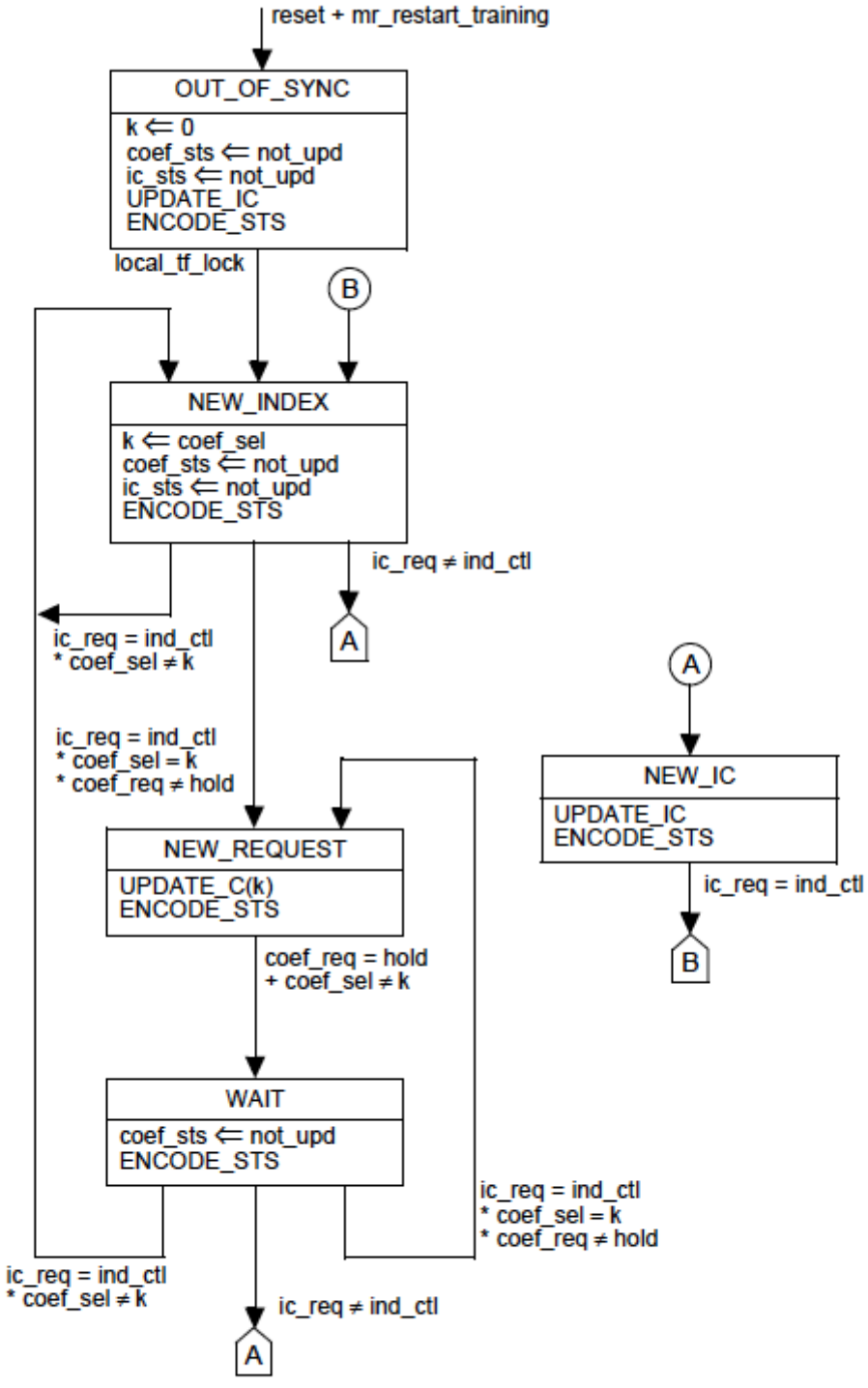


Figure 136–9—Coefficient update state diagram

Note: STATE_ALG_INIT, STATE_ALG_PAM4, STATE_ALG_PRESET, and STATE_ALG_DONE are internal proprietary link training algorithm states.

3.3.5 LT MSG

```
171406.708127, LT(S0),          MSG: LOCK=true, SYNC LOST=true, NEW_FRAME=true,␣
↪OVERRUN=false
```

LT MSG messages show log messages from a serdes lane of LT.

3.3.6 LT COEFF MSG

```
171406.708154, LT_COEF(S0),     FSM: (EVENT_RESET_DEASSERT) IDLE -> OUT_OF_SYNC
171406.708165, LT_COEF(S0),     MSG: Setting coeff c(-1) PRE1 to 0
171406.708175, LT_COEF(S0),     MSG: Setting coeff c(0)  MAIN to 68
171406.708185, LT_COEF(S0),     MSG: Setting coeff c(1)  POST to 0
171406.708196, LT_COEF(S0),     MSG: Setting coeff c(-2) PRE2 to 0
171406.708206, LT_COEF(S0),     MSG: Setting coeff c(-3) PRE3 to 0
171406.708217, LT(S0),          TX:  0x00000300, LOCKED=true, TRAINED=false
```

LT COEFF MSG messages show log messages of coefficient change of a serdes lane from LT.

3.3.7 LT TX & RX

```
171406.758221, LT(S0),          TX:  0x12000A00, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: IC 1   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.758319, LT_ALG0(S1),     FSM: (EVENT_ALG_SCAN_PRESET) STATE_ALG_PAM4 -> STATE_
↪ALG_PRESET
171406.758355, LT(S0),          RX:  0x02000B00, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: Upd    PAM_
↪MOD: PAM4
171406.758393, LT(S0),          TX:  0x02000A80, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.758410, LT(S1),          TX:  0x12000A00, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: IC 1   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.758461, LT(S0),          RX:  0x02000A80, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
```

The raw hex value of the transmitted and received LT test frames are shown first. Decoding of each field are shown after the raw value.

Note: The example above demonstrates a 4-way handshake of the link training transaction.

1. The port lane 0 requests the remote to use Preset 1 `C_REQ: Hold C_SEL: c(0) IC_REQ: IC 1 PAM_MOD: PAM4`
2. The remote confirms the update `C_ECH: c(0) C_STS: No upd IC_STS: Upd PAM_MOD: PAM4` without requesting any change.
3. The port tells the remote port to hold `C_REQ: Hold C_SEL: c(0) IC_REQ: INDV PAM_MOD: PAM4`
4. The remote port holds the change `C_ECH: c(0) C_STS: No upd IC_STS: No upd PAM_MOD: PAM4`

Control Field

The first line `C_REQ, C_SEL, IC_REQ, PAM_MOD` is control field information, defined in IEEE 802.3ck Table 162-9.

- `C_REQ`: Coefficient request
- `C_SEL`: Coefficient select
- `IC_REQ`, Initial condition request
- `PAM_MOD`, Modulation and precoding request

Status Field

The second line `C_ECH, C_STS, IC_STS, PAM_MOD` is status information, defined in IEEE 802.3ck Table 162-10.

- `C_ECH`: Coefficient select echo
- `C_STS`: Coefficient status
- `IC_STS`, Initial condition status
- `PAM_MOD`, Modulation and precoding status

3.3.8 Full Example

A complete log example is shown below.

```
Last login: Sun Mar 17 12:35:57 on ttys014
leonardyu@HRW-L000058 ~ % ssh 2342@localhost -p 22622
Hello 2342, welcome to Xena OpenAutomation ANLT Utility server (2.2.0)

xoa-utils > anlt log --read -f 20240317_1.log
171406.514179, LT(S0),          FSM: (EVENT_RESET_DEASSERT) IDLE -> INITIALIZE
171406.515194, LT(S1),          FSM: (EVENT_RESET_DEASSERT) IDLE -> INITIALIZE
171406.515279, LT(S0),          FSM: (EVENT_RESET_DEASSERT) IDLE -> INITIALIZE
171406.515291, LT(S1),          FSM: (EVENT_RESET_DEASSERT) IDLE -> INITIALIZE
171406.515335, ANEG,            FSM: (EVENT_INIT_DONE) IDLE -> WAIT_ANEG_ENABLE
171406.515382, ANEG,            FSM: (EVENT_AUTONEG_DISABLE) WAIT_ANEG_ENABLE -> AN_
↪GOOD_CHECK
171406.515404, LT(S0),          FSM: (EVENT_TRAINING_ENABLE) INITIALIZE -> START_
↪DELAY1
171406.515426, LT(S1),          FSM: (EVENT_TRAINING_ENABLE) INITIALIZE -> START_
```

(continues on next page)

IEEE Std 802.3ck-2022
IEEE Standard for Ethernet—Amendment 4: Physical Layer Specifications and Management Parameters for 100 Gb/s, 200 Gb/s, and 400 Gb/s Electrical Interfaces Based on 100 Gb/s Signaling

Table 162–9—Control field structure

Bit(s)	Name	Description
15:14	Reserved	Transmit as 0, ignore on receipt
13:11	Initial condition request	13 12 11 1 1 1 = Reserved 1 0 1 = Reserved 0 1 1 = Preset 5 0 0 1 = Preset 4 1 1 0 = Preset 3 1 0 0 = Preset 2 0 1 0 = Preset 1 0 0 0 = Individual coefficient control
10	Reserved	Transmit as 0, ignore on receipt
9:8	Modulation and precoding request	9 8 1 1 = PAM4 with precoding 1 0 = PAM4 0 1 = Reserved 0 0 = PAM2
7:5	Reserved	Transmit as 0, ignore on receipt
4:2	Coefficient select	4 3 2 1 0 0 = Reserved 1 0 1 = $c(-3)$ 1 1 0 = $c(-2)$ 1 1 1 = $c(-1)$ 0 0 0 = $c(0)$ 0 0 1 = $c(1)$ 0 1 x = Reserved
1:0	Coefficient request	1 0 1 1 = No equalization 1 0 = Decrement 0 1 = Increment 0 0 = Hold

IEEE Std 802.3ck-2022
IEEE Standard for Ethernet—Amendment 4: Physical Layer Specifications and Management Parameters for 100 Gb/s, 200 Gb/s, and 400 Gb/s Electrical Interfaces Based on 100 Gb/s Signaling

Table 162–10—Status field structure

Bit(s)	Name	Description
15	Receiver ready	1 =Training is complete and the receiver is ready for data 0 =Request for training to continue
14:12	Reserved	Transmit as 0, ignore on receipt
11:10	Modulation and precoding status	11 10 1 1 = PAM4 with precoding 1 0 = PAM4 0 1 = Reserved 0 0 = PAM2
9	Receiver frame lock	1 = Frame boundaries identified 0 = Frame boundaries not identified
8	Initial condition status	1 = Updated 0 = Not updated
7	Parity	Even parity bit
6	Reserved	Transmit as 0, ignore on receipt
5:3	Coefficient select echo	5 4 3 1 0 1 = $c(-3)$ 1 1 0 = $c(-2)$ 1 1 1 = $c(-1)$ 0 0 0 = $c(0)$ 0 0 1 = $c(1)$
2:0	Coefficient status	2 1 0 1 1 1 = Reserved 1 1 0 = Coefficient at limit and equalization limit 1 0 1 = Reserved 1 0 0 = Equalization limit 0 1 1 = Coefficient not supported 0 1 0 = Coefficient at limit 0 0 1 = Updated 0 0 0 = Not updated

(continued from previous page)

```
↪DELAY1
171406.518225, LT(S0),          FSM: (EVENT_RESET_DEASSERT) IDLE -> INITIALIZE
171406.519238, LT(S1),          FSM: (EVENT_RESET_DEASSERT) IDLE -> INITIALIZE
171406.519323, LT(S0),          FSM: (EVENT_RESET_DEASSERT) IDLE -> INITIALIZE
171406.519336, LT(S1),          FSM: (EVENT_RESET_DEASSERT) IDLE -> INITIALIZE
171406.519359, ANEG,            FSM: (EVENT_INIT_DONE) IDLE -> WAIT_ANEG_ENABLE
171406.519385, ANEG,            FSM: (EVENT_AUTONEG_ENABLE) WAIT_ANEG_ENABLE ->␣
↪TRANSMIT_DISABLE
171406.519396, ANEG,            MSG: Setting coeff c(-1) PRE1 to 0
171406.519407, ANEG,            MSG: Setting coeff c(0)  MAIN to 68
171406.519417, ANEG,            MSG: Setting coeff c(1)  POST to 0
171406.519428, ANEG,            MSG: Setting coeff c(-2) PRE2 to 0
171406.519438, ANEG,            MSG: Setting coeff c(-3) PRE3 to 0
171406.519452, ANEG,            MSG: TRANSMIT_DISABLE - ANEG restart
171406.519462, ANEG,            TX:  0x004000198001, base page, NP:1, ACK:0, RF:0,␣
↪TN:25, EN:0, C:0
                                     FEC:[], ABILITY:['200GBASE_KR2_CR2']
171406.519481, ANEG,            RX:  0x004000198001, base page, NP:1, ACK:0, RF:0,␣
↪TN:25, EN:0, C:0
                                     FEC:[], ABILITY:['200GBASE_KR2_CR2']
171406.586888, ANEG,            FSM: (EVENT_BREAK_LINK_TIMER_DONE) TRANSMIT_DISABLE ->
↪ ABILITY_DETECT
171406.586905, ANEG,            MSG: SYNC=true, SYNC LOST=false, NEW_PAGE=true
171406.586917, ANEG,            TX:  0x004000198001, base page, NP:1, ACK:0, RF:0,␣
↪TN:25, EN:0, C:0
                                     FEC:[], ABILITY:['200GBASE_KR2_CR2']
171406.586935, ANEG,            RX:  0x004000198001, base page, NP:1, ACK:0, RF:0,␣
↪TN:25, EN:0, C:0
                                     FEC:[], ABILITY:['200GBASE_KR2_CR2']
171406.586984, ANEG,            FSM: (EVENT_ABILITY_MATCH_NONCE) ABILITY_DETECT ->␣
↪TRANSMIT_DISABLE
171406.586995, ANEG,            MSG: Setting coeff c(-1) PRE1 to 0
171406.587005, ANEG,            MSG: Setting coeff c(0)  MAIN to 68
171406.587015, ANEG,            MSG: Setting coeff c(1)  POST to 0
171406.587026, ANEG,            MSG: Setting coeff c(-2) PRE2 to 0
171406.587036, ANEG,            MSG: Setting coeff c(-3) PRE3 to 0
171406.587050, ANEG,            MSG: TRANSMIT_DISABLE - ANEG restart
171406.587059, ANEG,            TX:  0x004000078001, base page, NP:1, ACK:0, RF:0,␣
↪TN:7, EN:0, C:0
                                     FEC:[], ABILITY:['200GBASE_KR2_CR2']
171406.654488, ANEG,            FSM: (EVENT_BREAK_LINK_TIMER_DONE) TRANSMIT_DISABLE ->
↪ ABILITY_DETECT
171406.654505, ANEG,            MSG: SYNC=false, SYNC LOST=true, NEW_PAGE=true
171406.654517, ANEG,            TX:  0x004000078001, base page, NP:1, ACK:0, RF:0,␣
↪TN:7, EN:0, C:0
                                     FEC:[], ABILITY:['200GBASE_KR2_CR2']
171406.654535, ANEG,            RX:  0x004000198001, base page, NP:1, ACK:0, RF:0,␣
↪TN:25, EN:0, C:0
                                     FEC:[], ABILITY:['200GBASE_KR2_CR2']
171406.654566, ANEG,            MSG: SYNC=true, SYNC LOST=true, NEW_PAGE=true
171406.654579, ANEG,            RX:  0x0040001B8001, base page, NP:1, ACK:0, RF:0,␣
↪TN:27, EN:0, C:0
                                     FEC:[], ABILITY:['200GBASE_KR2_CR2']
171406.654612, ANEG,            MSG: SYNC=true, SYNC LOST=false, NEW_PAGE=true
171406.654641, ANEG,            FSM: (EVENT_ABILITY_MATCH_N_NONCE) ABILITY_DETECT ->␣
↪ACKNOWLEDGE_DETECT
```

(continues on next page)

(continued from previous page)

```
171406.654653, ANEG,            TX:  0x00400007C361, base page, NP:1, ACK:1, RF:0, TN:7, EN:27, C:0
                                     FEC:[], ABILITY:['200GBASE_KR2_CR2']
171406.654707, ANEG,            RX:  0x0040001BC0E1, base page, NP:1, ACK:1, RF:0, TN:27, EN:7, C:0
                                     FEC:[], ABILITY:['200GBASE_KR2_CR2']
171406.654739, ANEG,            FSM: (EVENT_ACKNOWLEDGE_DETECT) ACKNOWLEDGE_DETECT -> COMPLETE_ACKNOWLEDGE
171406.654806, ANEG,            FSM: (EVENT_ACK_NP) COMPLETE_ACKNOWLEDGE -> NEXT_PAGE_WAIT
171406.654818, ANEG,            TX:  0x04DF0353A805, next page, NP:1, ACK:0, MP:1, ACK2:0, T:1
                                     Formatted message:
                                     Value:0x0005, Msg:OUI Tagged: 0x6a737c (preliminary)
171406.654837, ANEG,            RX:  0x04DF0353A805, next page, NP:1, ACK:0, MP:1, ACK2:0, T:1
                                     Formatted message:
                                     Value:0x0005, Msg:OUI Tagged: 0x6a737c (preliminary)
171406.654889, ANEG,            FSM: (EVENT_NEXT_PAGE) NEXT_PAGE_WAIT -> ACKNOWLEDGE_DETECT
171406.654901, ANEG,            TX:  0x04DF0353E805, next page, NP:1, ACK:1, MP:1, ACK2:0, T:1
                                     Formatted message:
                                     Value:0x0005, Msg:OUI Tagged: 0x6a737c (preliminary)
171406.654939, ANEG,            RX:  0x04DF0353E805, next page, NP:1, ACK:1, MP:1, ACK2:0, T:1
                                     Formatted message:
                                     Value:0x0005, Msg:OUI Tagged: 0x6a737c (preliminary)
171406.654973, ANEG,            FSM: (EVENT_ACKNOWLEDGE_DETECT) ACKNOWLEDGE_DETECT -> COMPLETE_ACKNOWLEDGE
171406.655022, ANEG,            FSM: (EVENT_ACK_NP) COMPLETE_ACKNOWLEDGE -> NEXT_PAGE_WAIT
171406.655034, ANEG,            TX:  0x000000000203, next page, NP:0, ACK:0, MP:0, ACK2:0, T:0
                                     Un-formatted message:
                                     Value:0x0203, Msg:OUI Tagged: 0x6a737d (final) Ethernet Technology Consortium
                                     FEC:[], ABILITY:[]
171406.655049, ANEG,            RX:  0x000000000203, next page, NP:0, ACK:0, MP:0, ACK2:0, T:0
                                     Un-formatted message:
                                     Value:0x0203, Msg:OUI Tagged: 0x6a737d (final) Ethernet Technology Consortium
                                     FEC:[], ABILITY:[]
171406.656373, ANEG,            FSM: (EVENT_NEXT_PAGE) NEXT_PAGE_WAIT -> ACKNOWLEDGE_DETECT
171406.656386, ANEG,            TX:  0x000000004203, next page, NP:0, ACK:1, MP:0, ACK2:0, T:0
                                     Un-formatted message:
                                     Value:0x0203, Msg:OUI Tagged: 0x6a737d (final) Ethernet Technology Consortium
                                     FEC:[], ABILITY:[]
```

(continues on next page)

(continued from previous page)

```
171406.658021, ANEG,            RX:  0x000000004203, next page, NP:0, ACK:1, MP:0, ACK2:0, T:0
                                     Un-formatted message:
                                     Value:0x0203, Msg:OUI Tagged: 0x6a737d (final) Ethernet Technology Consortium
                                     FEC:[], ABILITY:[]
171406.658050, ANEG,            FSM: (EVENT_ACKNOWLEDGE_DETECT) ACKNOWLEDGE_DETECT -> COMPLETE_ACKNOWLEDGE
171406.658080, ANEG,            FSM: (EVENT_ACK_N_NP) COMPLETE_ACKNOWLEDGE -> AN_GOOD_CHECK
171406.658103, LT(S0),          FSM: (EVENT_TRAINING_ENABLE) INITIALIZE -> START_DELAY1
171406.658125, LT(S1),          FSM: (EVENT_TRAINING_ENABLE) INITIALIZE -> START_DELAY1
171406.683096, LT(S0),          FSM: (EVENT_WAIT_TIMER_DONE) START_DELAY1 -> START_DELAY2
171406.683127, LT(S1),          FSM: (EVENT_WAIT_TIMER_DONE) START_DELAY1 -> START_DELAY2
171406.708103, LT(S0),          FSM: (EVENT_WAIT_TIMER_DONE) START_DELAY2 -> SEND_TF
171406.708115, LT(S0),          TX:  0x00000000, LOCKED=false, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_MOD: PAM2
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_MOD: PAM2
171406.708127, LT(S0),          MSG: LOCK=true, SYNC LOST=true, NEW_FRAME=true, OVERRUN=false
171406.708143, LT(S0),          RX:  0x00000000, LOCKED=false, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_MOD: PAM2
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_MOD: PAM2
171406.708154, LT_COEF(S0),     FSM: (EVENT_RESET_DEASSERT) IDLE -> OUT_OF_SYNC
171406.708165, LT_COEF(S0),     MSG: Setting coeff c(-1) PRE1 to 0
171406.708175, LT_COEF(S0),     MSG: Setting coeff c(0)  MAIN to 68
171406.708185, LT_COEF(S0),     MSG: Setting coeff c(1)  POST to 0
171406.708196, LT_COEF(S0),     MSG: Setting coeff c(-2) PRE2 to 0
171406.708206, LT_COEF(S0),     MSG: Setting coeff c(-3) PRE3 to 0
171406.708217, LT(S0),          TX:  0x00000300, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_MOD: PAM2
                                     C_ECH: c(0)   C_STS: No upd IC_STS: Upd    PAM_MOD: PAM2
171406.708230, LT_ALG0(S0),     FSM: (EVENT_RESET_DEASSERT) IDLE -> STATE_ALG_INIT
171406.708240, LT_ALG1(S0),     FSM: (EVENT_RESET_DEASSERT) IDLE -> STATE_ALG_INIT
171406.708251, LT(S1),          FSM: (EVENT_WAIT_TIMER_DONE) START_DELAY2 -> SEND_TF
171406.708263, LT(S1),          TX:  0x00000000, LOCKED=false, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_MOD: PAM2
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_MOD: PAM2
171406.708275, LT(S1),          MSG: LOCK=true, SYNC LOST=true, NEW_FRAME=true, OVERRUN=true
171406.708291, LT(S1),          RX:  0x00000000, LOCKED=false, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_MOD: PAM2
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
```

(continues on next page)

(continued from previous page)

```
↪MOD: PAM2
171406.708303, LT_COEF(S1),     FSM: (EVENT_RESET_DEASSERT) IDLE -> OUT_OF_SYNC
171406.708313, LT_COEF(S1),     MSG: Setting coeff c(-1) PRE1 to 0
171406.708324, LT_COEF(S1),     MSG: Setting coeff c(0)  MAIN to 68
171406.708334, LT_COEF(S1),     MSG: Setting coeff c(1)  POST to 0
171406.708345, LT_COEF(S1),     MSG: Setting coeff c(-2) PRE2 to 0
171406.708355, LT_COEF(S1),     MSG: Setting coeff c(-3) PRE3 to 0
171406.708366, LT(S1),          TX:  0x00000300, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM2
                                     C_ECH: c(0)   C_STS: No upd IC_STS: Upd    PAM_
↪MOD: PAM2
171406.708379, LT_ALG0(S1),     FSM: (EVENT_RESET_DEASSERT) IDLE -> STATE_ALG_INIT
171406.708390, LT_ALG1(S1),     FSM: (EVENT_RESET_DEASSERT) IDLE -> STATE_ALG_INIT
171406.708415, LT(S0),          RX:  0x00000300, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM2
                                     C_ECH: c(0)   C_STS: No upd IC_STS: Upd    PAM_
↪MOD: PAM2
171406.708428, LT_COEF(S0),     FSM: (EVENT_LOCAL_TF_LOCK) OUT_OF_SYNC -> NEW_INDEX
171406.708439, LT(S0),          TX:  0x00000280, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM2
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM2
171406.708454, LT(S1),          MSG: LOCK=true, SYNC LOST=false, NEW_FRAME=true,␣
↪OVERRUN=false
171406.708466, LT(S1),          RX:  0x00000180, LOCKED=false, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM2
                                     C_ECH: c(0)   C_STS: No upd IC_STS: Upd    PAM_
↪MOD: PAM2
171406.708478, LT_COEF(S1),     FSM: (EVENT_LOCAL_TF_LOCK) OUT_OF_SYNC -> NEW_INDEX
171406.708490, LT(S1),          TX:  0x00000280, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM2
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM2
171406.708517, LT(S0),          FSM: (EVENT_TEST_FRAME_LOCK) SEND_TF -> TRAIN_LOCAL
171406.708529, LT(S0),          RX:  0x00000280, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM2
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM2
171406.708552, LT_ALG0(S0),     FSM: (EVENT_ALG_SWITCH_PAM4) STATE_ALG_INIT -> STATE_
↪ALG_PAM4
171406.708575, LT(S1),          RX:  0x00000280, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM2
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM2
171406.708607, LT(S1),          FSM: (EVENT_TEST_FRAME_LOCK) SEND_TF -> TRAIN_LOCAL
171406.708629, LT_ALG0(S1),     FSM: (EVENT_ALG_SWITCH_PAM4) STATE_ALG_INIT -> STATE_
↪ALG_PAM4
171406.708665, LT(S0),          TX:  0x02000200, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
```

(continues on next page)

(continued from previous page)

```
↪MOD: PAM4
                                    C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM2
171406.708682, LT(S1),          RX:  0x02000200, LOCKED=true, TRAINED=false
                                    C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                    C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM2
171406.708697, LT(S1),          TX:  0x00000A00, LOCKED=true, TRAINED=false
                                    C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM2
                                    C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.708726, LT(S0),          RX:  0x02000200, LOCKED=true, TRAINED=false
                                    C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                    C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM2
171406.708741, LT(S0),          TX:  0x02000A80, LOCKED=true, TRAINED=false
                                    C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                    C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.712900, LT(S0),          RX:  0x02000A80, LOCKED=true, TRAINED=false
                                    C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                    C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.713344, LT(S1),          TX:  0x02000A80, LOCKED=true, TRAINED=false
                                    C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                    C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.713418, LT(S1),          RX:  0x02000A80, LOCKED=true, TRAINED=false
                                    C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                    C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.758160, LT_ALG0(S0),     FSM: (EVENT_ALG_SCAN_PRESET) STATE_ALG_PRESET ->␣
↪STATE_ALG_PRESET
171406.758221, LT(S0),          TX:  0x12000A00, LOCKED=true, TRAINED=false
                                    C_REQ: Hold   C_SEL: c(0)   IC_REQ: IC 1   PAM_
↪MOD: PAM4
                                    C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.758319, LT_ALG0(S1),     FSM: (EVENT_ALG_SCAN_PRESET) STATE_ALG_PAM4 -> STATE_
↪ALG_PRESET
171406.758355, LT(S0),          RX:  0x02000B00, LOCKED=true, TRAINED=false
                                    C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                    C_ECH: c(0)   C_STS: No upd IC_STS: Upd    PAM_
↪MOD: PAM4
171406.758393, LT(S0),          TX:  0x02000A80, LOCKED=true, TRAINED=false
                                    C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                    C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
```

(continues on next page)

(continued from previous page)

```
↪MOD: PAM4
171406.758410, LT(S1),          TX:  0x12000A00, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: IC 1   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.758461, LT(S0),          RX:  0x02000A80, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.758561, LT(S1),          RX:  0x02000B00, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: Upd    PAM_
↪MOD: PAM4
171406.758597, LT(S1),          RX:  0x12000B80, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: IC 1   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: Upd    PAM_
↪MOD: PAM4
171406.758611, LT_COEF(S1),     FSM: (EVENT_NEW_IC) NEW_INDEX -> NEW_IC
171406.758622, LT_COEF(S1),     MSG: Setting coeff c(-1) PRE1 to 0
171406.758633, LT_COEF(S1),     MSG: Setting coeff c(0)  MAIN to 68
171406.758643, LT_COEF(S1),     MSG: Setting coeff c(1)  POST to 0
171406.758653, LT_COEF(S1),     MSG: Setting coeff c(-2) PRE2 to 0
171406.758664, LT_COEF(S1),     MSG: Setting coeff c(-3) PRE3 to 0
171406.758675, LT(S1),          TX:  0x02000B00, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: Upd    PAM_
↪MOD: PAM4
171406.758748, LT(S1),          RX:  0x12000A00, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: IC 1   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.758784, LT(S1),          RX:  0x02000A80, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.758798, LT_COEF(S1),     FSM: (EVENT_NEW_REQ) NEW_IC -> NEW_INDEX
171406.758809, LT(S1),          TX:  0x02000A80, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.783254, LT(S0),          RX:  0x12000A00, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: IC 1   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.783269, LT_COEF(S0),     FSM: (EVENT_NEW_IC) NEW_INDEX -> NEW_IC
171406.783280, LT_COEF(S0),     MSG: Setting coeff c(-1) PRE1 to 0
171406.783290, LT_COEF(S0),     MSG: Setting coeff c(0)  MAIN to 68
```

(continues on next page)

(continued from previous page)

```
171406.783300, LT_COEF(S0),     MSG: Setting coeff c(1)  POST to 0
171406.783311, LT_COEF(S0),     MSG: Setting coeff c(-2) PRE2 to 0
171406.783321, LT_COEF(S0),     MSG: Setting coeff c(-3) PRE3 to 0
171406.783332, LT(S0),          TX:  0x02000B00, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: Upd    PAM_
↪MOD: PAM4
171406.783440, LT(S0),          RX:  0x02000A80, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.783454, LT_COEF(S0),     FSM: (EVENT_NEW_REQ) NEW_IC -> NEW_INDEX
171406.783466, LT(S0),          TX:  0x02000A80, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.808169, LT_ALG0(S0),     FSM: (EVENT_ALG_NEXT) STATE_ALG_PRESET -> STATE_ALG_
↪PRESET
171406.808230, LT(S0),          TX:  0x22000A00, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: IC 2   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.808365, LT(S0),          RX:  0x02000B00, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: Upd    PAM_
↪MOD: PAM4
171406.808415, LT_ALG0(S1),     FSM: (EVENT_ALG_NEXT) STATE_ALG_PRESET -> STATE_ALG_
↪PRESET
171406.808451, LT(S0),          TX:  0x02000A80, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.808468, LT(S1),          RX:  0x22000A00, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: IC 2   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.808483, LT_COEF(S1),     FSM: (EVENT_NEW_IC) NEW_INDEX -> NEW_IC
171406.808494, LT_COEF(S1),     MSG: Setting coeff c(-1) PRE1 to 0
171406.808504, LT_COEF(S1),     MSG: Setting coeff c(0)  MAIN to 42
171406.808514, LT_COEF(S1),     MSG: Setting coeff c(1)  POST to 0
171406.808524, LT_COEF(S1),     MSG: Setting coeff c(-2) PRE2 to 0
171406.808535, LT_COEF(S1),     MSG: Setting coeff c(-3) PRE3 to 0
171406.808545, LT(S1),          TX:  0x02000B00, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: Upd    PAM_
↪MOD: PAM4
171406.808576, LT(S0),          RX:  0x02000A80, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
```

(continues on next page)

(continued from previous page)

```
↪MOD: PAM4
                                    C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.808595, LT(S1),          TX:  0x22000B80, LOCKED=true, TRAINED=false
                                    C_REQ: Hold   C_SEL: c(0)   IC_REQ: IC 2   PAM_
↪MOD: PAM4
                                    C_ECH: c(0)   C_STS: No upd IC_STS: Upd    PAM_
↪MOD: PAM4
171406.808711, LT(S1),          RX:  0x02000B00, LOCKED=true, TRAINED=false
                                    C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                    C_ECH: c(0)   C_STS: No upd IC_STS: Upd    PAM_
↪MOD: PAM4
171406.808725, LT_COEF(S1),     FSM: (EVENT_NEW_REQ) NEW_IC -> NEW_INDEX
171406.808737, LT(S1),          TX:  0x22000A00, LOCKED=true, TRAINED=false
                                    C_REQ: Hold   C_SEL: c(0)   IC_REQ: IC 2   PAM_
↪MOD: PAM4
                                    C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.808771, LT(S1),          TX:  0x02000A80, LOCKED=true, TRAINED=false
                                    C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                    C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.808865, LT(S1),          RX:  0x02000A80, LOCKED=true, TRAINED=false
                                    C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                    C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.858203, LT_ALG0(S0),     FSM: (EVENT_ALG_NEXT) STATE_ALG_PRESET -> STATE_ALG_
↪PRESET
171406.858264, LT(S0),          TX:  0x32000A80, LOCKED=true, TRAINED=false
                                    C_REQ: Hold   C_SEL: c(0)   IC_REQ: IC 3   PAM_
↪MOD: PAM4
                                    C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.858298, LT(S0),          RX:  0x22000A00, LOCKED=true, TRAINED=false
                                    C_REQ: Hold   C_SEL: c(0)   IC_REQ: IC 2   PAM_
↪MOD: PAM4
                                    C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.858313, LT_COEF(S0),     FSM: (EVENT_NEW_IC) NEW_INDEX -> NEW_IC
171406.858324, LT_COEF(S0),     MSG: Setting coeff c(-1) PRE1 to 0
171406.858334, LT_COEF(S0),     MSG: Setting coeff c(0)  MAIN to 42
171406.858345, LT_COEF(S0),     MSG: Setting coeff c(1)  POST to 0
171406.858355, LT_COEF(S0),     MSG: Setting coeff c(-2) PRE2 to 0
171406.858365, LT_COEF(S0),     MSG: Setting coeff c(-3) PRE3 to 0
171406.858376, LT(S0),          TX:  0x32000B00, LOCKED=true, TRAINED=false
                                    C_REQ: Hold   C_SEL: c(0)   IC_REQ: IC 3   PAM_
↪MOD: PAM4
                                    C_ECH: c(0)   C_STS: No upd IC_STS: Upd    PAM_
↪MOD: PAM4
171406.858411, LT(S0),          RX:  0x22000B80, LOCKED=true, TRAINED=false
                                    C_REQ: Hold   C_SEL: c(0)   IC_REQ: IC 2   PAM_
↪MOD: PAM4
                                    C_ECH: c(0)   C_STS: No upd IC_STS: Upd    PAM_
```

(continues on next page)

(continued from previous page)

```
↪MOD: PAM4
171406.858461, LT_ALG0(S1),     FSM: (EVENT_ALG_NEXT) STATE_ALG_PRESET -> STATE_ALG_
↪PRESET
171406.858497, LT(S0),          TX:  0x02000B00, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: Upd    PAM_
↪MOD: PAM4
171406.858532, LT(S0),          RX:  0x02000B00, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: Upd    PAM_
↪MOD: PAM4
171406.858546, LT_COEF(S0),     FSM: (EVENT_NEW_REQ) NEW_IC -> NEW_INDEX
171406.858558, LT(S0),          TX:  0x02000A80, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.858574, LT(S1),          RX:  0x32000A80, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: IC 3   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.858589, LT_COEF(S1),     FSM: (EVENT_NEW_IC) NEW_INDEX -> NEW_IC
171406.858600, LT_COEF(S1),     MSG: Setting coeff c(-1) PRE1 to 8
171406.858610, LT_COEF(S1),     MSG: Setting coeff c(0)  MAIN to 52
171406.858621, LT_COEF(S1),     MSG: Setting coeff c(1)  POST to 0
171406.858631, LT_COEF(S1),     MSG: Setting coeff c(-2) PRE2 to 0
171406.858641, LT_COEF(S1),     MSG: Setting coeff c(-3) PRE3 to 0
171406.858652, LT(S1),          TX:  0x32000B00, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: IC 3   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: Upd    PAM_
↪MOD: PAM4
171406.858683, LT(S0),          RX:  0x02000A80, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.858783, LT(S1),          RX:  0x32000B00, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: IC 3   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: Upd    PAM_
↪MOD: PAM4
171406.858819, LT(S1),          RX:  0x02000B00, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: Upd    PAM_
↪MOD: PAM4
171406.858833, LT_COEF(S1),     FSM: (EVENT_NEW_REQ) NEW_IC -> NEW_INDEX
171406.858844, LT(S1),          TX:  0x02000A80, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
```

(continues on next page)

(continued from previous page)

```
171406.858918, LT(S1),          RX:  0x02000A80, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.908314, LT(S0),          RX:  0x32000A80, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: IC 3   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.908329, LT_COEF(S0),     FSM: (EVENT_NEW_IC) NEW_INDEX -> NEW_IC
171406.908340, LT_COEF(S0),     MSG: Setting coeff c(-1) PRE1 to 8
171406.908350, LT_COEF(S0),     MSG: Setting coeff c(0)  MAIN to 52
171406.908361, LT_COEF(S0),     MSG: Setting coeff c(1)  POST to 0
171406.908371, LT_COEF(S0),     MSG: Setting coeff c(-2) PRE2 to 0
171406.908381, LT_COEF(S0),     MSG: Setting coeff c(-3) PRE3 to 0
171406.908392, LT(S0),          TX:  0x02000B00, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: Upd    PAM_
↪MOD: PAM4
171406.908541, LT(S0),          RX:  0x02000A80, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.908555, LT_COEF(S0),     FSM: (EVENT_NEW_REQ) NEW_IC -> NEW_INDEX
171406.908567, LT(S0),          TX:  0x02000A80, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.908583, LT(S1),          RX:  0x0A000A00, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: IC 4   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.908598, LT_COEF(S1),     FSM: (EVENT_NEW_IC) NEW_INDEX -> NEW_IC
171406.908609, LT_COEF(S1),     MSG: Setting coeff c(-1) PRE1 to 18
171406.908619, LT_COEF(S1),     MSG: Setting coeff c(0)  MAIN to 52
171406.908630, LT_COEF(S1),     MSG: Setting coeff c(1)  POST to 0
171406.908640, LT_COEF(S1),     MSG: Setting coeff c(-2) PRE2 to 5
171406.908650, LT_COEF(S1),     MSG: Setting coeff c(-3) PRE3 to 0
171406.908661, LT(S1),          TX:  0x02000B00, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: Upd    PAM_
↪MOD: PAM4
171406.908736, LT(S1),          RX:  0x02000A80, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.908750, LT_COEF(S1),     FSM: (EVENT_NEW_REQ) NEW_IC -> NEW_INDEX
171406.908761, LT(S1),          TX:  0x02000A80, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
```

(continues on next page)

(continued from previous page)

```
↪MOD: PAM4
                                        C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.933241, LT_ALG0(S0),     FSM: (EVENT_ALG_NEXT) STATE_ALG_PRESET -> STATE_ALG_
↪PRESET
171406.933302, LT(S0),          TX:  0x0A000A00, LOCKED=true, TRAINED=false
                                        C_REQ: Hold   C_SEL: c(0)   IC_REQ: IC 4   PAM_
↪MOD: PAM4
                                        C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.933437, LT(S0),          RX:  0x02000B00, LOCKED=true, TRAINED=false
                                        C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                        C_ECH: c(0)   C_STS: No upd IC_STS: Upd    PAM_
↪MOD: PAM4
171406.933486, LT_ALG0(S1),     FSM: (EVENT_ALG_NEXT) STATE_ALG_PRESET -> STATE_ALG_
↪PRESET
171406.933524, LT(S0),          TX:  0x02000A80, LOCKED=true, TRAINED=false
                                        C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                        C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.933562, LT(S1),          TX:  0x0A000A00, LOCKED=true, TRAINED=false
                                        C_REQ: Hold   C_SEL: c(0)   IC_REQ: IC 4   PAM_
↪MOD: PAM4
                                        C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.933593, LT(S0),          RX:  0x02000A80, LOCKED=true, TRAINED=false
                                        C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                        C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.933693, LT(S1),          RX:  0x02000B00, LOCKED=true, TRAINED=false
                                        C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                        C_ECH: c(0)   C_STS: No upd IC_STS: Upd    PAM_
↪MOD: PAM4
171406.933729, LT(S1),          TX:  0x02000A80, LOCKED=true, TRAINED=false
                                        C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                        C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.933803, LT(S1),          RX:  0x02000A80, LOCKED=true, TRAINED=false
                                        C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                        C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.958569, LT(S1),          RX:  0x32000A80, LOCKED=true, TRAINED=false
                                        C_REQ: Hold   C_SEL: c(0)   IC_REQ: IC 3   PAM_
↪MOD: PAM4
                                        C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.958584, LT_COEF(S1),     FSM: (EVENT_NEW_IC) NEW_INDEX -> NEW_IC
171406.958595, LT_COEF(S1),     MSG: Setting coeff c(-1) PRE1 to 8
171406.958605, LT_COEF(S1),     MSG: Setting coeff c(0)  MAIN to 52
171406.958615, LT_COEF(S1),     MSG: Setting coeff c(1)  POST to 0
```

(continues on next page)

(continued from previous page)

```
171406.958626, LT_COEF(S1),     MSG: Setting coeff c(-2) PRE2 to 0
171406.958636, LT_COEF(S1),     MSG: Setting coeff c(-3) PRE3 to 0
171406.958647, LT(S1),          TX:  0x02000B00, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: Upd    PAM_
↪MOD: PAM4
171406.958741, LT(S1),          RX:  0x02000A80, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.958755, LT_COEF(S1),     FSM: (EVENT_NEW_REQ) NEW_IC -> NEW_INDEX
171406.958766, LT(S1),          TX:  0x02000A80, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.983247, LT_ALG0(S0),     FSM: (EVENT_ALG_NEXT) STATE_ALG_PRESET -> STATE_ALG_
↪PRESET
171406.983308, LT(S0),          TX:  0x32000A80, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: IC 3   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.983343, LT(S0),          RX:  0x0A000A00, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: IC 4   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.983358, LT_COEF(S0),     FSM: (EVENT_NEW_IC) NEW_INDEX -> NEW_IC
171406.983369, LT_COEF(S0),     MSG: Setting coeff c(-1) PRE1 to 18
171406.983379, LT_COEF(S0),     MSG: Setting coeff c(0)  MAIN to 52
171406.983390, LT_COEF(S0),     MSG: Setting coeff c(1)  POST to 0
171406.983400, LT_COEF(S0),     MSG: Setting coeff c(-2) PRE2 to 5
171406.983410, LT_COEF(S0),     MSG: Setting coeff c(-3) PRE3 to 0
171406.983421, LT(S0),          TX:  0x32000B00, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: IC 3   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: Upd    PAM_
↪MOD: PAM4
171406.983456, LT(S0),          RX:  0x0A000B80, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: IC 4   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: Upd    PAM_
↪MOD: PAM4
171406.983506, LT_ALG0(S1),     FSM: (EVENT_ALG_NEXT) STATE_ALG_PRESET -> STATE_ALG_
↪PRESET
171406.983542, LT(S0),          RX:  0x02000B00, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                     C_ECH: c(0)   C_STS: No upd IC_STS: Upd    PAM_
↪MOD: PAM4
171406.983556, LT_COEF(S0),     FSM: (EVENT_NEW_REQ) NEW_IC -> NEW_INDEX
171406.983568, LT(S0),          TX:  0x02000A80, LOCKED=true, TRAINED=false
                                     C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
```

(continues on next page)

(continued from previous page)

```
↪MOD: PAM4
                                    C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.983606, LT(S1),          TX:  0x32000A80, LOCKED=true, TRAINED=false
                                    C_REQ: Hold   C_SEL: c(0)   IC_REQ: IC 3   PAM_
↪MOD: PAM4
                                    C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.983657, LT(S0),          RX:  0x02000A80, LOCKED=true, TRAINED=false
                                    C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                    C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.983737, LT(S1),          RX:  0x02000B00, LOCKED=true, TRAINED=false
                                    C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                    C_ECH: c(0)   C_STS: No upd IC_STS: Upd    PAM_
↪MOD: PAM4
171406.983773, LT(S1),          TX:  0x02000A80, LOCKED=true, TRAINED=false
                                    C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                    C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171406.983867, LT(S1),          RX:  0x02000A80, LOCKED=true, TRAINED=false
                                    C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                    C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171407.033343, LT(S0),          RX:  0x32000A80, LOCKED=true, TRAINED=false
                                    C_REQ: Hold   C_SEL: c(0)   IC_REQ: IC 3   PAM_
↪MOD: PAM4
                                    C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171407.033358, LT_COEF(S0),     FSM: (EVENT_NEW_IC) NEW_INDEX -> NEW_IC
171407.033369, LT_COEF(S0),     MSG: Setting coeff c(-1) PRE1 to 8
171407.033379, LT_COEF(S0),     MSG: Setting coeff c(0)  MAIN to 52
171407.033390, LT_COEF(S0),     MSG: Setting coeff c(1)  POST to 0
171407.033400, LT_COEF(S0),     MSG: Setting coeff c(-2) PRE2 to 0
171407.033410, LT_COEF(S0),     MSG: Setting coeff c(-3) PRE3 to 0
171407.033422, LT(S0),          TX:  0x02000B00, LOCKED=true, TRAINED=false
                                    C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                    C_ECH: c(0)   C_STS: No upd IC_STS: Upd    PAM_
↪MOD: PAM4
171407.033510, LT(S0),          RX:  0x02000A80, LOCKED=true, TRAINED=false
                                    C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                    C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171407.033524, LT_COEF(S0),     FSM: (EVENT_NEW_REQ) NEW_IC -> NEW_INDEX
171407.033536, LT(S0),          TX:  0x02000A80, LOCKED=true, TRAINED=false
                                    C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                    C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171407.033653, LT(S1),          RX:  0x02008A00, LOCKED=true, TRAINED=true
```

(continues on next page)

(continued from previous page)

```
                                        C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                        C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171407.058271, LT_ALG0(S0),     FSM: (EVENT_ALG_DONE) STATE_ALG_PRESET -> STATE_ALG_
↪DONE
171407.058292, LT_ALG0(S0),     MSG: {'cmds': [{'ber': ['56', '68'], 'cmd': 'SET␣
↪PRESET_1', 'flags': ['DONE', 'LOCK'], 'prbs': [{'bits': 2081396800, 'errors': 68,
↪'result': '3.267e-08'}], 'result': 'success'}, {'ber': ['20745', '199'], 'cmd':
↪'SET PRESET_2', 'flags': ['DONE', 'LOCK'], 'prbs': [{'bits': 2082624640, 'errors':␣
↪199, 'result': '9.555e-08'}], 'result': 'success'}, {'ber': ['1114', '1', '0'], 'cmd
↪': 'SET PRESET_3', 'flags': ['DONE', 'LOCK'], 'prbs': [{'bits': 4164585280, 'errors
↪': 1, 'result': '2.401e-10'}], 'result': 'success'}, {'ber': ['11397', '1775'], 'cmd
↪': 'SET PRESET_4', 'flags': ['DONE', 'LOCK'], 'prbs': [{'bits': 2081435520, 'errors
↪': 1775, 'result': '8.528e-07'}], 'result': 'success'}, {'ber': ['20857', '0', '3'],
↪  'cmd': 'SET PRESET_3', 'flags': ['DONE', 'LOCK'], 'prbs': [{'bits': 4164583040,
↪'errors': 3, 'result': '7.204e-10'}], 'result': 'success'}, {'cmd': 'LOCAL_TRAINED'}
↪]}
171407.058340, LT(S0),          FSM: (EVENT_LOCAL_TRAINED) TRAIN_LOCAL -> TRAIN_REMOTE
171407.058352, LT(S0),          TX:  0x02008A00, LOCKED=true, TRAINED=true
                                        C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                        C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171407.058520, LT_ALG0(S1),     FSM: (EVENT_ALG_DONE) STATE_ALG_PRESET -> STATE_ALG_
↪DONE
171407.058541, LT_ALG0(S1),     MSG: {'cmds': [{'ber': ['193', '131'], 'cmd': 'SET␣
↪PRESET_1', 'flags': ['DONE', 'LOCK'], 'prbs': [{'bits': 2084357760, 'errors': 131,
↪'result': '6.285e-08'}], 'result': 'success'}, {'ber': ['7456', '872'], 'cmd': 'SET␣
↪PRESET_2', 'flags': ['DONE', 'LOCK'], 'prbs': [{'bits': 2084303360, 'errors': 872,
↪'result': '4.184e-07'}], 'result': 'success'}, {'ber': ['443', '2', '2'], 'cmd':
↪'SET PRESET_3', 'flags': ['DONE', 'LOCK'], 'prbs': [{'bits': 4163505280, 'errors':␣
↪4, 'result': '9.607e-10'}], 'result': 'success'}, {'ber': ['15475', '214'], 'cmd':
↪'SET PRESET_4', 'flags': ['DONE', 'LOCK'], 'prbs': [{'bits': 2081495680, 'errors':␣
↪214, 'result': '1.028e-07'}], 'result': 'success'}, {'ber': ['1166', '3', '4'], 'cmd
↪': 'SET PRESET_3', 'flags': ['DONE', 'LOCK'], 'prbs': [{'bits': 4163755520, 'errors
↪': 7, 'result': '1.681e-09'}], 'result': 'success'}, {'cmd': 'LOCAL_TRAINED'}]}
171407.058588, LT(S1),          FSM: (EVENT_LOCAL_TRAINED) TRAIN_LOCAL -> TRAIN_REMOTE
171407.058601, LT(S1),          TX:  0x02008A00, LOCKED=true, TRAINED=true
                                        C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                        C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171407.058634, LT(S1),          FSM: (EVENT_REMOTE_RX_READY) TRAIN_REMOTE -> LINK_
↪READY
171407.058765, LT(S1),          FSM: (EVENT_WAIT_TIMER_DONE) LINK_READY -> SEND_DATA
171407.058776, LT_COEF(S1),     FSM: (XFSM_EVENT_SELF) NEW_INDEX -> IDLE
171407.058787, LT_ALG0(S1),     FSM: (XFSM_EVENT_SELF) STATE_ALG_DONE -> IDLE
171407.058797, LT_ALG1(S1),     FSM: (XFSM_EVENT_SELF) STATE_ALG_INIT -> IDLE
171407.108387, LT(S0),          RX:  0x02008A00, LOCKED=true, TRAINED=true
                                        C_REQ: Hold   C_SEL: c(0)   IC_REQ: INDV   PAM_
↪MOD: PAM4
                                        C_ECH: c(0)   C_STS: No upd IC_STS: No upd PAM_
↪MOD: PAM4
171407.108419, LT(S0),          FSM: (EVENT_REMOTE_RX_READY) TRAIN_REMOTE -> LINK_
↪READY
```

(continues on next page)

(continued from previous page)

```
171407.108555, LT(S0),          FSM: (EVENT_WAIT_TIMER_DONE) LINK_READY -> SEND_DATA
171407.108567, LT_COEF(S0),     FSM: (XFSM_EVENT_SELF) NEW_INDEX -> IDLE
171407.108577, LT_ALG0(S0),     FSM: (XFSM_EVENT_SELF) STATE_ALG_DONE -> IDLE
171407.108588, LT_ALG1(S0),     FSM: (XFSM_EVENT_SELF) STATE_ALG_INIT -> IDLE
171407.108615, ANEG,            FSM: (EVENT_LINK_HCD_OK) AN_GOOD_CHECK -> AN_GOOD
171415.054007, LT(S0),          FSM: (EVENT_RESET_DEASSERT) IDLE -> INITIALIZE
171415.054970, LT(S1),          FSM: (EVENT_RESET_DEASSERT) IDLE -> INITIALIZE
171415.055053, LT(S0),          FSM: (EVENT_RESET_DEASSERT) IDLE -> INITIALIZE
171415.055065, LT(S1),          FSM: (EVENT_RESET_DEASSERT) IDLE -> INITIALIZE
171415.055111, ANEG,            FSM: (EVENT_INIT_DONE) IDLE -> WAIT_ANEG_ENABLE
171415.055158, ANEG,            FSM: (EVENT_AUTONEG_ENABLE) WAIT_ANEG_ENABLE ->␣
↪TRANSMIT_DISABLE
171415.055169, ANEG,            MSG: Setting coeff c(-1) PRE1 to 0
171415.055179, ANEG,            MSG: Setting coeff c(0)  MAIN to 68
171415.055190, ANEG,            MSG: Setting coeff c(1)  POST to 0
171415.055200, ANEG,            MSG: Setting coeff c(-2) PRE2 to 0
171415.055210, ANEG,            MSG: Setting coeff c(-3) PRE3 to 0
171415.055224, ANEG,            MSG: TRANSMIT_DISABLE - ANEG restart
171415.055233, ANEG,            TX:  0x004000158001, base page, NP:1, ACK:0, RF:0,␣
↪TN:21, EN:0, C:0
                                     FEC:[], ABILITY:['200GBASE_KR2_CR2']
171415.057863, LT(S0),          FSM: (EVENT_RESET_DEASSERT) IDLE -> INITIALIZE
171415.058993, LT(S1),          FSM: (EVENT_RESET_DEASSERT) IDLE -> INITIALIZE
171415.059100, LT(S0),          FSM: (EVENT_RESET_DEASSERT) IDLE -> INITIALIZE
171415.059112, LT(S1),          FSM: (EVENT_RESET_DEASSERT) IDLE -> INITIALIZE
171415.059136, ANEG,            FSM: (EVENT_INIT_DONE) IDLE -> WAIT_ANEG_ENABLE
171415.059162, ANEG,            FSM: (EVENT_AUTONEG_DISABLE) WAIT_ANEG_ENABLE -> AN_
↪GOOD_CHECK
171415.059175, LT(S0),          FSM: (EVENT_TRAINING_DISABLE) INITIALIZE -> SEND_DATA
171415.059187, LT(S1),          FSM: (EVENT_TRAINING_DISABLE) INITIALIZE -> SEND_DATA
171415.059212, ANEG,            FSM: (EVENT_LINK_HCD_OK) AN_GOOD_CHECK -> AN_GOOD
xoa-utils >
```

CHAPTER

FOUR

COMMAND REFERENCE

4.1 Summary

Table 1: Management Command Summary

Command	Description	Example
connect	Connect to tester	`connect 10.10.10.10 yourname`
port	Reserve and switch port	`port 0/0 | port 0/0 --reset`
ports	List ports	`ports | ports --all`
module-config	Set module media and port config	`module-config 0 osfp800 8 100g`
exit	Exit the session	`exit`

Table 2: AN/LT Command Summary

Command	Description	Example
anlt start	Apply and start AN/LT on the port	`anlt start`
anlt stop	Stop AN/LT on the port	`anlt stop`
anlt log	Show AN/LT protocol trace log and save to a file	`anlt log --filename mylog.log`
anlt log	Read saved log file	`anlt log --read -f saved_mylog.log`
anlt autorestart	Control AN/LT autorestart	`anlt autorestart --link-down --lt-fail`
anlt status	Show AN/LT status of the local port	`anlt status`
anlt strict	Enable/disable ANLt strict mode	`anlt strict --on`
anlt logctrl	Control what should be logged in ANLT by xenaserver	`anlt logctrl -DALGPNmTcsZO`

Table 3: AN Command Summary

Command	Description	Example
an config	Configure AN of the local port	`an config --on --loopback`
an status	Show AN status of the local port	`an status`

Table 4: LT Command Summary

Command	Description	Example
lt config	Configure LT of the local port	`lt config --on --mode auto --preset0 ieee --timeout enable`
lt encoding	Request **remote port** to use the specified encoding on the specified serdes	`lt encoding 0 pam4`
lt preset	Request **remote port** to use the preset of the specified serdes	`lt preset 0 2`
lt inc	Request **remote port** to increase (+) its emphasis value by 1	`lt inc 0 main`
lt dec	Request **remote port** to decrease (-) its emphasis value by 1	`lt dec 0 main`
lt status	Show the link training status of the specified serdes of the local port	`lt status 0`
lt trained	Announce that the specified serdes is trained	`lt trained 0`
lt no-eq	Request **remote port** to turn off equalizer on its emphasis	`lt no-eq 0 main`
lt im	Set initial modulation for the specified serdes of the local port	`lt im 0 nrz`
lt alg	Set the link training algorithm for the specified serdes	`lt alg 0 alg0`
lt txtapget	Read the tap values of the specified serdes of the local port	`lt txtapget 0`
lt txtapset	Write the tap values of the specified serdes of the local port	`lt txtapset 0 1 3 4 60 1`
lt txtap-autotune	Auto tune the tap values of the specified serdes of the local port	`lt txtap-autotune 0`

4.2 Management Commands

4.2.1 connect

Description

Connect to a tester for the current session.

Synopsis

```
connect <DEVICE> <USERNAME>
[-p, --ports <value: text list>]
[--reset/--no-reset]
[--force/--no-force]
[-P, --password <value: text>]
[-t, --tcp_port <value: integer>]
```

Arguments

<DEVICE> (text)

Specifies the chassis address for connection.

You can specify the IP addresses in IPv4 format, or a host name, e.g. `10.10.10.10` or `demo.xenanetworks.com`

<USERNAME> (text)

Specifies the name of the user, e.g. `xoa` or `automation`

Options

`-p, --ports` (text list)

Specifies the ports on the specified device host, default to null.

Specify a port using the format slot/port, no space between. e.g. `--ports 0/0,0/1,0/2,0/3`.

If used, the context will switch to the first port in the list after the connection is established.

`--reset/--no-reset`

Removes all port configurations of the ports in *–ports* after reservation, default to `--reset`.

`--force/--no-force`

Breaks port locks established by another user, aka. force reservation, default to `--force`.

`-P, --password` (text)

The login password of the tester, default to `xena`.

`-t, --tcp` (int)

The TCP port number on the chassis for the client to establish a session, default to `22606`.

Examples

```
xoa-utils > connect 10.10.10.10 xoa

Tester  :       9999999
ConInfo :       10.10.10.102:22606
Username:       xoa

Port      Sync      Owner

xoa-utils[123456][port0/0] >
```

4.2.2 exit

Description

Exit the session by terminating port reservations, disconnecting from the chassis, releasing system resources, and removing the specified port configurations. This command works in all context.

Synopsis

```
exit
[--reset/--no-reset]
[--release/--no-release]
```

Arguments

Options

`--reset/--no-reset`

Removes all port configurations of the ports in `--ports` after reservation, default to `--reset`.

`--release/--no-release`

Determines whether the ports should be released before exiting, default to `--release`.

Examples

```
xoa-utils[123456][port0/2] > exit
```

4.2.3 port

Description

Reserve and switch port. If the port is not yet reserved, reserve the port. This command changes the working port and will stay in the same context.

Synopsis

```
port <PORT>
[--reset/--no-reset]
[--force/--no-force]
```

Arguments

<PORT> (text)

Specifies the port on the specified device host.

Specify a port using the format slot/port, e.g. 0/0

Options

`--reset/--no-reset`

Removes the port configurations, default to `--no-reset`.

`--force/--no-force`

Breaks port locks established by another user, aka. force reservation, default to `--force`.

Examples

```
xoa-utils[123456] > port 0/0
Port      Sync      Owner
*2/0      IN_SYNC   You

[ ACTUAL CONFIG ]
    Link recovery         : off
    Serdes count          : 1

    Auto-negotiation      : off (allow loopback: yes)
    Link training         : on (interactive) (preset0: standard tap values) (timeout:␣
↪disabled)
        Initial Mod.      : {'0': 'NRZ'}


[ SHADOW CONFIG ]
    Auto-negotiation      : off (allow loopback: no)
    Link training         : on (interactive) (preset0: standard tap values)
        Initial Mod.      : {'0': 'NRZ'}
```

4.2.4 ports

Description

List all the ports reserved by the current session. This command works in all context.

Synopsis

```
ports
[--all/--no-all]
```

Arguments

Options

`--all/--no-all`

Show all ports of the tester, default to `--no-all`

Examples

```
xoa-utils[123456][port0/0] > ports
Ports       Sync        Owner
*0/0        yes         You
```

```
xoa-utils[123456][port0/0] > ports --all
Port      Sync      Owner
*0/0      IN_SYNC   You
0/1       IN_SYNC   Others
6/0       NO_SYNC   Others
6/1       NO_SYNC   Others
```

4.2.5 module-config

Description

Set module's media configuration and port speed configuration.

Synopsis

```
module-config <MODULE> <MEDIA> <PORT_COUNT> <PORT_SPEED>
[--force/--no-force]
```

Arguments

<MODULE> (text)

Specifies the module on the specified device host.

Specify a module using the format slot, e.g. 0

<MEDIA> (text)

Specifies the media configuration type of the module.

Allowed values:

- *cfp4*
- *cxp*
- *sfp28*
- *qsfp28_nrz*
- *qsfp28_pam4*
- *qsfp56_pam4*
- *qsfpdd_pam4*
- *sfp56*
- *sfpdd*
- *sfp112*
- *qsfpdd_nrz*
- *cfp*
- *base_t1*
- *base_t1s*
- *qsfpdd800*
- *qsfp112*
- *osfp800*

<PORT_COUNT> (integer)

Specifies the port count of the module.

<PORT_SPEED> (text)

Specifies the port speed in Gbps of the module.

Allowed values:

- *10g*

- *25g*
- *50g*
- *100g*
- *200g*
- *400g*
- *800g*

Options

`--force/--no-force`

Breaks module locks established by another user and all the ports of the module, aka. force reservation, default to `--force`.

Examples

```
xoa-utils[123456] > module-config 0 qsfpdd800 8 100g
```

4.3 AN & LT Commands

4.3.1 ANLT Group

Commands for AN/LT.

anlt start

Description

Apply and start AN/LT to the working port.

Apply the shadow configuration from *an config*, *lt config*, and *lt im* to the working port, and then start AN/LT on the port.

Synopsis

```
anlt start
```

Arguments

Options

Examples

```
xoa-utils[123456][port0/0] > anlt start

xoa-utils[123456][port0/0] >
```

anlt stop

Description

Stop AN/LT to the working port.

Synopsis

```
anlt stop
```

Arguments

Options

Examples

```
xoa-utils[123456][port0/0] > anlt stop

xoa-utils[123456][port0/0] >
```

anlt autorestart

Description

Control AN/LT autorestart.

Synopsis

```
anlt autorestart
[--link-down/--no-link-down]
[--lt-fail/--no-lt-fail]
```

Arguments

Options

`--link-down/--no-link-down`

Should port enables AN+LT autorestart when a link down condition is detected, default to `--no-link-down`

`--lt-fail/--no-lt-fail`

Should port initiates the AN+LT restart process repeatedly when LT experiences failure until LT succeeds, default to `--no-lt-fail`.

Examples

```
xoa-utils[123456][port0/2] > anlt autorestart --link-down --lt-fail
```

anlt status

Description

Show auto-negotiation and link training actual and shadow configurations of the working port.

Synopsis

```
anlt status
```

Arguments

Options

Examples

```
xoa-utils[123456][port0/0] > anlt status

[ACTUAL CONFIG]
    Link recovery         : off
    Serdes count          : 1

    Auto-negotiation      : on (allow loopback: yes)
    Link training         : on (auto) (preset0: standard tap values) (timeout:␣
↪default)


[SHADOW CONFIG]
    Auto-negotiation      : on (allow loopback: no)
    Link training         : off (auto) (preset0: standard tap values)

xoa-utils[123456][port0/2] >
```

anlt log

Description

Show ANLT protocol trace log and save to a file.

To **quit** the continuous display mode, press `Control-z`.

Synopsis

```
anlt log
[-f, --filename <value: text>]
[-k, --keep <value: text>]
[-s, --serdes <value>]
[--read, <value: text>]
```

Arguments

Options

`-f, --filename` (text)

Specifies the filename for the log messages to be stored.

`--read -f` (text)

Specifies the filepath of the log file to display.

`-k, --keep` (text)

Specifies what types of log entries to keep, default to keep all.

Allowed values:

- *all*, to keep all.
- *an*. to keep autoneg only.
- *lt*, to keep lt only.

`-s, --serdes` (int list)

Specifies which serdes of LT logs to keep. If you don't know how many serdes serdes the port has, use *anlt log*, default to all serdes.

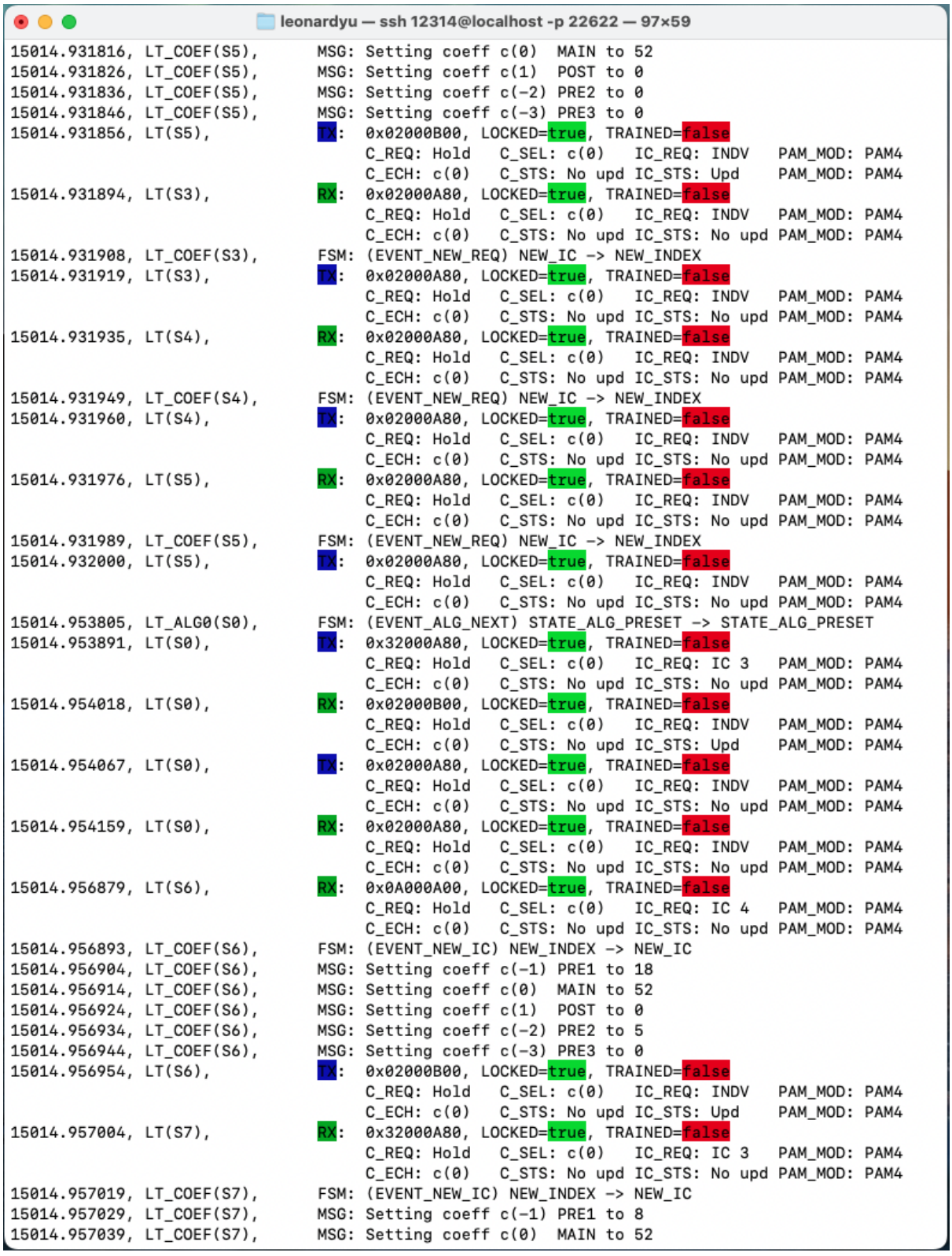
Examples

```
xoa-utils[123456][port0/2] > anlt log --filename mylog.log --keep lt --serdes 0
```

anlt logctrl

Description

Control what types of ANLT log messages are sent by xenaserver. This command is different from the `--keep` option of *anlt log*. `anlt log-ctrl` control the log message from its source, where `anlt_log` filters the messages for display output.


leonardyu — ssh 12314@localhost -p 22622 — 97×59
15014.931816, LT_COEF(S5), MSG: Setting coeff c(0) MAIN to 52
15014.931826, LT_COEF(S5), MSG: Setting coeff c(1) POST to 0
15014.931836, LT_COEF(S5), MSG: Setting coeff c(-2) PRE2 to 0
15014.931846, LT_COEF(S5), MSG: Setting coeff c(-3) PRE3 to 0
15014.931856, LT(S5), TX: 0x02000B00, LOCKED=true, TRAINED=false
C_REQ: Hold C_SEL: c(0) IC_REQ: INDV PAM_MOD: PAM4
C_ECH: c(0) C_STS: No upd IC_STS: Upd PAM_MOD: PAM4
15014.931894, LT(S3), RX: 0x02000A80, LOCKED=true, TRAINED=false
C_REQ: Hold C_SEL: c(0) IC_REQ: INDV PAM_MOD: PAM4
C_ECH: c(0) C_STS: No upd IC_STS: No upd PAM_MOD: PAM4
15014.931908, LT_COEF(S3), FSM: (EVENT_NEW_REQ) NEW_IC -> NEW_INDEX
15014.931919, LT(S3), TX: 0x02000A80, LOCKED=true, TRAINED=false
C_REQ: Hold C_SEL: c(0) IC_REQ: INDV PAM_MOD: PAM4
C_ECH: c(0) C_STS: No upd IC_STS: No upd PAM_MOD: PAM4
15014.931935, LT(S4), RX: 0x02000A80, LOCKED=true, TRAINED=false
C_REQ: Hold C_SEL: c(0) IC_REQ: INDV PAM_MOD: PAM4
C_ECH: c(0) C_STS: No upd IC_STS: No upd PAM_MOD: PAM4
15014.931949, LT_COEF(S4), FSM: (EVENT_NEW_REQ) NEW_IC -> NEW_INDEX
15014.931960, LT(S4), TX: 0x02000A80, LOCKED=true, TRAINED=false
C_REQ: Hold C_SEL: c(0) IC_REQ: INDV PAM_MOD: PAM4
C_ECH: c(0) C_STS: No upd IC_STS: No upd PAM_MOD: PAM4
15014.931976, LT(S5), RX: 0x02000A80, LOCKED=true, TRAINED=false
C_REQ: Hold C_SEL: c(0) IC_REQ: INDV PAM_MOD: PAM4
C_ECH: c(0) C_STS: No upd IC_STS: No upd PAM_MOD: PAM4
15014.931989, LT_COEF(S5), FSM: (EVENT_NEW_REQ) NEW_IC -> NEW_INDEX
15014.932000, LT(S5), TX: 0x02000A80, LOCKED=true, TRAINED=false
C_REQ: Hold C_SEL: c(0) IC_REQ: INDV PAM_MOD: PAM4
C_ECH: c(0) C_STS: No upd IC_STS: No upd PAM_MOD: PAM4
15014.953805, LT_ALG0(S0), FSM: (EVENT_ALG_NEXT) STATE_ALG_PRESET -> STATE_ALG_PRESET
15014.953891, LT(S0), TX: 0x32000A80, LOCKED=true, TRAINED=false
C_REQ: Hold C_SEL: c(0) IC_REQ: IC 3 PAM_MOD: PAM4
C_ECH: c(0) C_STS: No upd IC_STS: No upd PAM_MOD: PAM4
15014.954018, LT(S0), RX: 0x02000B00, LOCKED=true, TRAINED=false
C_REQ: Hold C_SEL: c(0) IC_REQ: INDV PAM_MOD: PAM4
C_ECH: c(0) C_STS: No upd IC_STS: Upd PAM_MOD: PAM4
15014.954067, LT(S0), TX: 0x02000A80, LOCKED=true, TRAINED=false
C_REQ: Hold C_SEL: c(0) IC_REQ: INDV PAM_MOD: PAM4
C_ECH: c(0) C_STS: No upd IC_STS: No upd PAM_MOD: PAM4
15014.954159, LT(S0), RX: 0x02000A80, LOCKED=true, TRAINED=false
C_REQ: Hold C_SEL: c(0) IC_REQ: INDV PAM_MOD: PAM4
C_ECH: c(0) C_STS: No upd IC_STS: No upd PAM_MOD: PAM4
15014.956879, LT(S6), RX: 0x0A000A00, LOCKED=true, TRAINED=false
C_REQ: Hold C_SEL: c(0) IC_REQ: IC 4 PAM_MOD: PAM4
C_ECH: c(0) C_STS: No upd IC_STS: No upd PAM_MOD: PAM4
15014.956893, LT_COEF(S6), FSM: (EVENT_NEW_IC) NEW_INDEX -> NEW_IC
15014.956904, LT_COEF(S6), MSG: Setting coeff c(-1) PRE1 to 18
15014.956914, LT_COEF(S6), MSG: Setting coeff c(0) MAIN to 52
15014.956924, LT_COEF(S6), MSG: Setting coeff c(1) POST to 0
15014.956934, LT_COEF(S6), MSG: Setting coeff c(-2) PRE2 to 5
15014.956944, LT_COEF(S6), MSG: Setting coeff c(-3) PRE3 to 0
15014.956954, LT(S6), TX: 0x02000B00, LOCKED=true, TRAINED=false
C_REQ: Hold C_SEL: c(0) IC_REQ: INDV PAM_MOD: PAM4
C_ECH: c(0) C_STS: No upd IC_STS: Upd PAM_MOD: PAM4
15014.957004, LT(S7), RX: 0x32000A80, LOCKED=true, TRAINED=false
C_REQ: Hold C_SEL: c(0) IC_REQ: IC 3 PAM_MOD: PAM4
C_ECH: c(0) C_STS: No upd IC_STS: No upd PAM_MOD: PAM4
15014.957019, LT_COEF(S7), FSM: (EVENT_NEW_IC) NEW_INDEX -> NEW_IC
15014.957029, LT_COEF(S7), MSG: Setting coeff c(-1) PRE1 to 8
15014.957039, LT_COEF(S7), MSG: Setting coeff c(0) MAIN to 52

Synopsis

```
anlt logctrl
[-D/-d, --debug/--no-debug]
[-A/-a, --an-trace/--no-an-trace]
[-L/-l, --lt-trace/--no-lt-trace]
[-G/-g, --alg-trace/--no-alg-trace]
[-P/-p, --fsm-port/--no-fsm-port]
[-N/-n, --fsm-an/--no-fsm-an]
[-M/-m, --fsm-an-stimuli/--no-fsm-an-stimuli]
[-T/-t, --fsm-lt/--no-fsm-lt]
[-C/-c, --fsm-lt-coeff/--no-fsm-lt-coeff]
[-S/-s, --fsm-lt-stimuli/--no-fsm-lt-stimuli]
[-Z/-z, --fsm-lt-alg0/--no-fsm-lt-alg0]
[-O/-o, --fsm-lt-algn1/--no-fsm-lt-algn1]
```

Arguments

Options

`-D/-d, --debug/--no-debug`

Debug log out, default to `--debug, -D`

`-A/-a, --an-trace/--no-an-trace`

Auto-negotiation trace output, default to –an-trace, -A

`-L/-l, --lt-trace/--no-lt-trace`

Link training algorithm trace, default to –lt-trace, -L

`-G/-g, --alg-trace/--no-alg-trace`

Link training algorithm trace output, default to –alg-trace, -G

`-P/-p, --fsm-port/--no-fsm-port`

Port state machine transitions output, default to –no-fsm-port, -p

`-N/-n, --fsm-an/--no-fsm-an`

Auto-negotiation state machine transitions, default to –fsm-an, -N

`-M/-m, --fsm-an-stimuli/--no-fsm-an-stimuli`

Auto-negotiation stimuli state machine transitions, default to –no-fsm-an-stimuli, -m

`-T/-t, --fsm-lt/--no-fsm-lt`

Link training state machine transitions, default to –fsm-lt, -T

`-C/-c, --fsm-lt-coeff/--no-fsm-lt-coeff`

Link training coefficient state machine transitions, default to –no-fsm-lt-coeff, -c

`-S/-s, --fsm-lt-stimuli/--no-fsm-lt-stimuli`

Link training stimuli state machine transitions, default to –no-fsm-lt-stimuli, -s

`-Z/-z, --fsm-lt-alg0/--no-fsm-lt-alg0`

Link training algorithm 0 state machine transitions, default to –fsm-lt-alg0, -Z

`-O/-o, --fsm-lt-algn1/--no-fsm-lt-algn1`

Link training algorithm -1 state machine transitions, default to –fsm-lt-algn1, -O

Examples

```
xoa-utils[123456][port0/2] > anlt logctrl
Port 0/2 log control:
    Type debug:             on
    Type AN trace:          on
    Type LT trace:          on
    Type ALG trace:         on
    Type FSM port:          on
    Type FSM AN:            on
    Type FSM AN Stimuli:    off
    Type FSM LT:            on
    Type FSM LT Coeff:      off
    Type FSM LT Stimuli:    off
    Type FSM LT ALG  0:     on
    Type FSM LT ALG -1:     on

xoa-utils[123456][port0/2] >
```

anlt strict

Description

Enable/disable ANLt strict mode. If enable, errored frames will be ignored.

Synopsis

```
anlt strict
[--on/--off]
```

Arguments

Options

`--on/--off`

Should enable ANLT strict mode, default to `--on`.

Examples

```
xoa-utils[123456][port0/2] > anlt strict --on
Port 0/2 ANLT strict mode: on

xoa-utils[123456][port0/2] >
```

4.3.2 AN Group

Commands for Auto-Negotiation.

an config

Important: This command only changes the local AN configuration state. To execute the configuration, you need to run *anlt start*, otherwise your changes will not take effect on the tester.

Description

Configure AN of the working port.

Synopsis

```
an config
[--on/--off]
[--loopback/--no-loopback]
```

Arguments

Options

`--on/--off`

Enable or disable auto-negotiation on the working port, default to `--on`.

`--loopback/--no-loopback`

Should loopback be allowed in auto-negotiation, default to `--no-loopback`.

Examples

Listing 1: Autoneg should be enabled and allow loopback

```
xoa-utils[123456][port0/2] > an config --on --loopback

AN configuration to be on port 2/0
[SHADOW CONFIG]
    Auto-negotiation      : on (allow loopback: yes)
    Link training         : off (auto) (preset0: standard tap values)

xoa-utils[123456][port0/2] >
```

an status

Description

Show the auto-negotiation status of the working port.

Synopsis

```
an status
```

Arguments

Options

Examples

```
xoa-utils[123456][port0/2] > an status

[AN STATUS]
    Mode                  : enabled
    Loopback              : allowed

    Duration              : 2,068,747 µs
    Successful runs       : 1
    Timeouts              : 0
    Loss of sync          : 0

    HCD                   : IEEE_800GBASE_CR8_KR8
    HCD negotiation fails : 0
    FEC result            : RS_FEC_KP
    FEC negotiation fails : 0

                               RX    TX
    Link codewords        :     2     1
    Next-page messages    :     0     0
    Unformatted pages     :     0     0

xoa-utils[123456][port0/2] >
```

4.3.3 LT Group

Commands for Link Training.

lt config

Important: This command only changes the local LT configuration state. To execute the configuration, you need to run *anlt start*, otherwise your changes will not take effect on the tester.

Description

Configure LT for the working port.

Synopsis

```
lt config
[--on/--off]
[-m, --mode <value: text>]
[--preset0 <value: text>]
[--timeout <value: text>]
```

Arguments

Options

`--on/--off`

Enable or disable link training on the working port, default to `--on`.

`-m, --mode` (text)

The mode for link training on the working port, default to `auto`.

Allowed values:

- *auto*: link training procedures are done by the port.
- *interactive*: link training procedures requires manual operation.

`--preset0` (text)

The preset0 mode, default to *standard* `.

Allowed values:

- *standard*: Use standard tap values as preset0
- *existing*: Use the existing tap values as preset0

`--timeout` (text)

Timeout mode, default to *enable* `.

Allowed values:

- *enable*: Enable timeout for LT auto
- *disable*: Disable timeout for LT auto

Examples

```
xoa-utils[123456][port0/2] > lt config --on --preset0=existing --mode=interactive --
↪timeout disable

LT configuration to be on port 2/0
[SHADOW CONFIG]
    Auto-negotiation      : off (allow loopback: no)
    Link training         : on (interactive) (preset0: existing tap values)

xoa-utils[123456][port0/2] >
```

lt im

Important: This command only changes the local LT configuration state. To execute the configuration, you need to run *anlt start*, otherwise your changes will not take effect on the tester.

Description

Set initial modulation for the specified serdes.

Synopsis

```
lt im <SERDES> <ENCODING>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

<ENCODING> (text)

Specifies the initial modulation.

Allowed values:

- *nrz*
- *pam4*
- *pam4pre*

Options

Examples

```
xoa-utils[123456][port0/2] > lt im 0 nrz

Initial modulation to be NRZ on Serdes 0
[SHADOW CONFIG]
    Auto-negotiation      : off (allow loopback: no)
    Link training         : on (interactive) (preset0: standard tap values)
        Initial Mod.      : {'0': 'NRZ'}

xoa-utils[123456][port0/2] >
```

lt alg

Important: This command only changes the local LT configuration state. To execute the configuration, you need to run *anlt start*, otherwise your changes will not take effect on the tester.

Description

Set the link training algorithm for the specified serdes.

Synopsis

```
lt alg <SERDES> <ALGORITHM>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

<ALGORITHM> (text)

Specifies the algorithm.

Allowed values:

- *alg0*
- *algn1*

Options

Examples

```
xoa-utils[123456][port0/2] > lt alg 0 alg0

Initial modulation to be NRZ on Serdes 0
[SHADOW CONFIG]
    Auto-negotiation      : off (allow loopback: no)
    Link training         : on (interactive) (preset0: standard tap values)
        Algorithm         : {'0': 'ALG0'}

xoa-utils[123456][port0/2] >
```

lt inc

Description

Request the remote link training partner to increase (+) its emphasis value by 1.

Synopsis

```
lt inc <SERDES> <EMPHASIS>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

<EMPHASIS> (text)

The emphasis (coefficient) of the link partner.

Allowed values:

- *pre3*
- *pre2*
- *pre*
- *main*
- *post*

Options

Examples

```
xoa-utils[123456][port0/2] > lt inc 0 main
Port 0/0: increase c(0) by 1 on Serdes 0 (COEFF_STS_UPDATED)

xoa-utils[123456][port0/2] >
```

lt dec

Description

Request the remote link training partner to decrease (-) its emphasis value by 1.

Synopsis

```
lt dec <SERDES> <EMPHASIS>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

<EMPHASIS> (text)

The emphasis (coefficient) of the link partner.

Allowed values:

- *pre3*
- *pre2*
- *pre*
- *main*
- *post*

Options

Examples

```
xoa-utils[123456][port0/2] > lt dec 0 main
Port 0/0: decrease c(0) by 1 on Serdes 0 (COEFF_STS_UPDATED)

xoa-utils[123456][port0/2] >
```

lt no-eq

Description

Request the remote link training turn off equalizer on its emphasis.

Synopsis

```
lt no-eq <SERDES> <EMPHASIS>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

<EMPHASIS> (text)

The emphasis (coefficient) of the link partner.

Allowed values:

- *pre3*
- *pre2*
- *pre*
- *main*
- *post*

Options

Examples

```
xoa-utils[123456][port0/2] > lt no-eq 0 main
Port 0/0: Turning off equalizer on c(0) on Serdes 0 (COEFF_STS_UPDATED)

xoa-utils[123456][port0/2] >
```

lt encoding

Description

Request the remote link training partner to use the specified encoding on the specified serdes.

Synopsis

```
lt encoding <SERDES> <ENCODING>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

<ENCODING> (text)

Specifies the encoding.

Allowed values:

- *nrz*
- *pam4*
- *pam4pre*

Options

Examples

```
xoa-utils[123456][port0/2] > lt encoding 0 pam4
Port 0/0: use PAM4 on Serdes 0 (SUCCESS)

xoa-utils[123456][port0/2] >
```

lt preset

Description

Request the remote link training partner to use the preset of the specified serdes.

Synopsis

```
lt preset <SERDES> <PRESET>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

<PRESET> (integer)

Specifies the preset index.

Allowed values: *1, 2, 3, 4, 5*

Options

Examples

```
xoa-utils[123456][port0/2] > lt preset 0 1
Port 0/0: use preset 0 on Serdes 0 (SUCCESS)

xoa-utils[123456][port0/2] >
```

lt trained

Description

Announce that the specified serdes is trained.

Synopsis

```
lt trained <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > lt trained 0
Port 0/0 requests: Serdes 0 is trained (SUCCESS)

xoa-utils[123456][port0/2] >
```

lt txtapget

Description

Read the tap values of the specified serdes of the local port.

Synopsis

```
lt txtapget <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > lt txtapget 0

Local Coefficient Serdes(0) :          c(-3)       c(-2)       c(-1)       c(0)      ␣
↪  c(1)
   Current level           :             0           0           0          42      ␣
↪     0

xoa-utils[123456][port0/2] >
```

lt txtapset

Description

Write the tap values of the specified serdes of the local port.

Synopsis

```
lt txtapset <SERDES> <PRE3> <PRE2> <PRE> <MAIN> <POST>
```

Arguments

<SERDES> (integer)

Specifies the serdes index.

<PRE3> (integer)

Specifies c(-3) value of the tap.

<PRE2> (integer)

Specifies c(-2) value of the tap.

<PRE> (integer)

Specifies c(-1) value of the tap.

<MAIN> (integer)

Specifies c(0) value of the tap.

<POST> (integer)

Specifies c(1) value of the tap.

Options

Examples

```
xoa-utils[123456][port0/2] > lt txtapset 5 1 6 5 80 0

Local Coefficient Serdes(5) :           c(-3)       c(-2)       c(-1)       c(0)
↪  c(1)
    Current level           :              1           6           5          80
↪     0

xoa-utils[123456][port0/2] >
```

lt txtap-autotune

Description

Auto tune the tap values of the specified serdes of the local port.

Synopsis

```
lt txtap-autotune <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > lt txtap-autotune 0

Local Coefficient Serdes(0) :           c(-3)       c(-2)       c(-1)       c(0)
↪  c(1)
    Current level           :              0           0           0          42
↪     0

xoa-utils[123456][port0/2] >
```

lt status

Description

Show the link training status of the specified serdes.

Synopsis

```
lt status <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > lt status 0

[LT STATUS]
    Mode              : on
    Status            : trained
    Failure           : no_failure

    Initial mod.      : nrz
    Preset0           : standard tap values

    Total bits        : 2,201,372,480
    Total err. bits   : 24
    BER               : 1.09e-08

    Duration          : 2,000,250 µs

    Lock lost         : 2
    Frame lock        : locked
    Remote frame lock : locked

    Frame errors      : 1
    Overrun errors    : 1

    Last IC received  : Preset 3
    Last IC sent      : Preset 3

    TX Coefficient              :          c(-3)       c(-2)       c(-1)        c(0) ␣
↪       c(1)
       Current level            :             0           0           1          44 ␣
↪          0
                                :         RX  TX      RX  TX      RX  TX      RX  TX ␣
↪    RX  TX
       + req                    :          0   0       0   0       2   2       1   1 ␣
```

(continues on next page)

(continued from previous page)

```
↪     0   0
       - req                    :          0   0       0   0       2   2       0   0 ␣
↪     0   0
       coeff/eq limit reached   :          0   0       0   0       0   0       0   0 ␣
↪     0   0
       eq limit reached         :          0   0       0   0       0   0       0   0 ␣
↪     0   0
       coeff not supported      :          0   0       0   0       0   0       0   0 ␣
↪     0   0
       coeff at limit           :          0   0       0   0       0   0       0   0 ␣
↪     0   0

xoa-utils[123456][port0/2] >
```

4.3.4 Debug Group

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

debug init

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Initialize debug

Synopsis

```
debug init <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index

Options

Examples

```
xoa-utils[123456][port0/2] > debug init 0

xoa-utils[123456][port0/2] >
```

debug serdes-reset

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug, reset the serdes.

Synopsis

```
debug serdes-reset <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug serdes-reset 0

xoa-utils[123456][port0/2] >
```

debug mode-get

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug mode-get

Synopsis

```
debug mode-get <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug mode-get 0

xoa-utils[123456][port0/2] >
```

debug mode-set

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug mode-set

Synopsis

```
debug mode-set <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug mode-set 0

xoa-utils[123456][port0/2] >
```

debug an-status

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug an-status

Synopsis

```
debug an-status <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug an-status 0

xoa-utils[123456][port0/2] >
```

debug an-tx-config-get

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug an-tx-config-get

Synopsis

```
debug an-tx-config-get <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug an-tx-config-get 0

xoa-utils[123456][port0/2] >
```

debug an-tx-config-set

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug an-tx-config-set

Synopsis

```
debug an-tx-config-set <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug an-tx-config-set 0

xoa-utils[123456][port0/2] >
```

debug an-rx-config-get

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug an-rx-config-get

Synopsis

```
debug an-rx-config-get <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug an-rx-config-get 0

xoa-utils[123456][port0/2] >
```

debug an-rx-config-set

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug an-rx-config-set

Synopsis

```
debug an-rx-config-set <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug an-rx-config-set 0

xoa-utils[123456][port0/2] >
```

debug an-rx-dme-mv-range-get

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug an-rx-dme-mv-range-get

Synopsis

```
debug an-rx-dme-mv-range-get <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug an-rx-dme-mv-range-get 0

xoa-utils[123456][port0/2] >
```

debug an-rx-dme-mv-range-set

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug an-rx-dme-mv-range-set

Synopsis

```
debug an-rx-dme-mv-range-set <SERDES> <VALUE>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

<VALUE> (integer)

Options

Examples

```
xoa-utils[123456][port0/2] > debug an-rx-dme-mv-range-set 0 1234

xoa-utils[123456][port0/2] >
```

debug an-rx-dme-bit-range-get

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug an-rx-dme-bit-range-get

Synopsis

```
debug an-rx-dme-bit-range-get <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug an-rx-dme-bit-range-get 0

xoa-utils[123456][port0/2] >
```

debug an-rx-dme-bit-range-set

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug an-rx-dme-bit-range-set

Synopsis

```
debug an-rx-dme-bit-range-set <SERDES> <VALUE>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

<VALUE> (integer)

Options

Examples

```
xoa-utils[123456][port0/2] > debug an-rx-dme-bit-range-set 0 1234

xoa-utils[123456][port0/2] >
```

debug lt-tx-config-get

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug lt-tx-config-get

Synopsis

```
debug lt-tx-config-get <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug lt-tx-config-get 0

xoa-utils[123456][port0/2] >
```

debug lt-tx-config-set

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug lt-tx-config-set

Synopsis

```
debug lt-tx-config-set <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug lt-tx-config-set 0

xoa-utils[123456][port0/2] >
```

debug lt-rx-config-get

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug lt-rx-config-get

Synopsis

```
debug lt-rx-config-get <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug lt-rx-config-get 0

xoa-utils[123456][port0/2] >
```

debug lt-rx-config-set

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug lt-rx-config-set

Synopsis

```
debug lt-rx-config-set <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug lt-rx-config-set 0

xoa-utils[123456][port0/2] >
```

debug lt-tx-tf-get

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug lt-tx-tf-get

Synopsis

```
debug lt-tx-tf-get <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug lt-tx-tf-get 0

xoa-utils[123456][port0/2] >
```

debug lt-tx-tf-set

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug lt-tx-tf-set

Synopsis

```
debug lt-tx-tf-set <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug lt-tx-tf-set 0

xoa-utils[123456][port0/2] >
```

debug lt-rx-tf-get

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug lt-rx-tf-get

Synopsis

```
debug lt-rx-tf-get <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug lt-rx-tf-get 0

xoa-utils[123456][port0/2] >
```

debug lt-status

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug lt-status

Synopsis

```
debug lt-status <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug lt-status 0

xoa-utils[123456][port0/2] >
```

debug lt-rx-error-stat0-get

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug lt-rx-error-stat0-get

Synopsis

```
debug lt-rx-error-stat0-get <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug lt-rx-error-stat0-get 0

xoa-utils[123456][port0/2] >
```

debug lt-rx-error-stat1-get

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug lt-rx-error-stat1-get

Synopsis

```
debug lt-rx-error-stat1-get <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug lt-rx-error-stat1-get 0

xoa-utils[123456][port0/2] >
```

debug xla-config-get

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug xla-config-get

Synopsis

```
debug xla-config-get <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug xla-config-get 0

xoa-utils[123456][port0/2] >
```

debug xla-config-set

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug xla-config-set

Synopsis

```
debug xla-config-set <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug xla-config-set 0

xoa-utils[123456][port0/2] >
```

debug xla-trig-mask-get

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug xla-trig-mask-get

Synopsis

```
debug xla-trig-mask-get <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug xla-trig-mask-get 0

xoa-utils[123456][port0/2] >
```

debug xla-trig-mask-set

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug xla-trig-mask-set

Synopsis

```
debug xla-trig-mask-set <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug xla-trig-mask-set 0

xoa-utils[123456][port0/2] >
```

debug xla-status-get

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug xla-status-get

Synopsis

```
debug xla-status-get <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug xla-status-get 0

xoa-utils[123456][port0/2] >
```

debug xla-rd-addr-get

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug xla-rd-addr-get

Synopsis

```
debug xla-rd-addr-get <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug xla-rd-addr-get 0

xoa-utils[123456][port0/2] >
```

debug xla-rd-addr-set

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug xla-rd-addr-set

Synopsis

```
debug xla-rd-addr-set <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug xla-rd-addr-set 0

xoa-utils[123456][port0/2] >
```

debug xla-rd-page-get

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug xla-rd-page-get

Synopsis

```
debug xla-rd-page-get <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug xla-rd-page-get 0

xoa-utils[123456][port0/2] >
```

debug xla-rd-page-set

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug xla-rd-page-set

Synopsis

```
debug xla-rd-page-set <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug xla-rd-page-set 0

xoa-utils[123456][port0/2] >
```

debug xla-rd-data-get

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug xla-rd-data-get

Synopsis

```
debug xla-rd-data-get <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug xla-rd-data-get 0

xoa-utils[123456][port0/2] >
```

debug lt-prbs

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug lt-prbs

Synopsis

```
debug lt-prbs <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug lt-prbs 0

xoa-utils[123456][port0/2] >
```

debug xla-dump

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug xla-dump

Synopsis

```
debug xla-dump <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

Examples

```
xoa-utils[123456][port0/2] > debug xla-dump 0

xoa-utils[123456][port0/2] >
```

debug xla-dump-ctrl

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Control XLA auto dump

Synopsis

```
debug xla-dump-ctrl
[--on/--off]
```

Arguments

Options

Examples

```
xoa-utils[123456][port0/2] > anlt xla-dump-ctrl
```

debug xla-trig-n-dump

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug xla-trig-n-dump

Synopsis

```
debug xla-trig-n-dump <SERDES>
```

Arguments

<SERDES> (integer)

Specifies the transceiver serdes index.

Options

`--mask, -m`

Mask, default to `0x00000FF0`.

`--window-offset, -o`

Window offset, default to `0x0080`.

`--trigger-select, -s`

Trigger select, default to `0x0001`.

`--filename, -f`

Trigger select, default to `xla_dump.csv`.

Examples

```
xoa-utils[123456][port0/2] > debug xla-trig-n-dump

xoa-utils[123456][port0/2] >
```

debug px-get

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug px-get

Synopsis

```
debug px-get <PAGE_ADDRESS> <REG_ADDRESS>
```

Arguments

<PAGE_ADDRESS> (integer)

<REG_ADDRESS> (string)

Options

Examples

```
xoa-utils[123456][port0/2] > debug px-get 2000 0x2f505
```

debug px-set

Important: To debug on a serdes, you must always use *debug init* command prior to all the other debug commands.

Description

Debug px-set

Synopsis

```
debug px-set <PAGE_ADDRESS> <REG_ADDRESS> <VALUE>
```

Arguments

<PAGE_ADDRESS> (integer)

<REG_ADDRESS> (string)

<VALUE> (string)

Options

Examples

```
xoa-utils[123456][port0/2] > debug px-set 2000 0x2f50 0x0101
```

CHAPTER

FIVE

GLOSSARY OF TERMS

AN
Auto-Negotiation

ANLT
Auto-Negotiation and Link Training

API
Application Programming Interface.

LT
Link Training

Test Resource
Test chassis, test module, and test port, both hardware and virtual are referred to as test resources. A user must have the ownership of a test resource before be able to perform testing.

TGA
Traffic Generation and Analysis.

Xena ANLT Utility
Xena Auto-Negotiation and Link Training Utility

XOA
Xena OpenAutomation

INDEX