



# Pseudo Random Binary Sequence

- [Pseudo Random Binary Sequence, on page 1](#)

## Pseudo Random Binary Sequence

The Pseudo Random Binary Sequence (PRBS) feature enables data integrity checks between NCS1014 trunk and client links without generating client traffic.

To ensure that the traffic is error-free during link bring up without relying on the peer port, you must enable the PRBS feature on both the transmitting and receiving ports of your NCS 1014 trunk. The transmitting trunk port creates a bit pattern and sends it to the peer NCS 1014 device. The device then confirms if the sent bit pattern is received. The 1.2T card supports PRBS on the ODU4 controller.

You can configure PRBS on the NCS 1014 trunk and client ports of a 1.2T, 2.4T, and 2.4TX cards.

- **Source mode** — The NCS 1014 at trunk port generates PRBS signal on the line continuously as per the configured PRBS pattern.
- **Sink mode** — The NCS 1014 at trunk port gets locked to the ingress signal according to the configured pattern, analyzes and reports the errors.
- **Source-Sink mode** — The NCS 1014 at trunk port acts as both the PRBS transmitter and receiver, that is, it generates PRBS signal as per the configured pattern, and also gets locked to the ingress signal with the same pattern, and reports the errors.

## Configure Pseudo Random Binary Sequence

You can configure PRBS on a coherentDSP or Ethernet controller of a 1.2T, 2.4T, or 2.4TX card. PRBS can also be configured on a ODU controller of the 1.2T card. Before enabling PRBS, the secondary admin state of the controllers must be set to *maintenance*.

### Configure PRBS on CoherentDSP Controller

To configure PRBS on the trunk port of the coherentDSP controller of a 1.2T, 2.4T, or 2.4TX card, use the following configuration commands in the configuration mode:

```
controller coherentDSP R/S/I/P
```

```
secondary-admin-state maintenance
```

**prbs mode {source | sink | source-sink} pattern {pn31 | pn23 | pn15 | pn7}**

### Example to Configure PRBS on CoherentDSP Controller:

Use the following sample configuration to configure PRBS on trunk ports of a coherentDSP controller:

```
RP/0/RP0/CPU0:ios(config)#controller CoherentDSP 0/0/0/7
RP/0/RP0/CPU0:ios(config-CoDSP)#secondary-admin-state maintenance
RP/0/RP0/CPU0:ios(config-CoDSP)#prbs mode source-sink pattern pn15
RP/0/RP0/CPU0:ios(config-CoDSP)#commit
Wed Nov 15 18:11:55.450 UTC
```

### Configure PRBS on ODU Controller

You can configure PRBS in the following combinations for an ODU controller on a 1.2T card. The client ports need to have a physical loopback in all the combinations.

- Near End client and Near End trunk ODU4
- Near End client and Far End client ODU4
- Near End client and Far End trunk ODU4
- Near End trunk and Far End trunk ODU4

To enable PRBS on ODU4 controller's trunk ports of a 1.2T card, use these commands in configuration mode:

**controller odu4 R/S/I/P**

**secondary-admin-state maintenance**

**opu prbs mode {source | sink | source-sink} pattern {pn31 | pn23 | pn15 | pn7}**

Example to Configure PRBS on ODU Controller:

Following is an example to configure PRBS on client ports of an ODU controller of a 1.2T card:

```
RP/0/RP0/CPU0:ios(config)#controller odu4 0/0/0/4
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#prbs mode source-sink pattern pn23
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#sec-admin-state maintenance
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#commit
```

### Configure PRBS on Ethernet Controller

To configure PRBS on the client ports of an Ethernet controller of the 2.4T and 2.4TX cards, use the following commands in the configuration mode:

**controller {fourHundredGigECtrlr} R/S/I/P**

**secondary-admin-state maintenance**

**opu prbs mode {source | sink | source-sink} pattern {pn31 | pn23 }**

Example to Configure PRBS on Ethernet Controller:

Following is an example to configure PRBS on client ports of an Ethernet controller of a 2.4T card:

```
RP/0/RP0/CPU0:ios(config)#controller FourHundredGigECtrlr 0/3/0/1
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#prbs mode source-sink pattern pn31
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#sec-admin-state maintenance
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#commit
```

## Verifying PRBS

You can monitor the status of Pseudo Random Binary Sequence (PRBS) using the following command:

```
show controllers coherentDSP | ODU4 | fourHundredGigEctrlr R/S/I/P prbs-details
```

### Example to view PRBS details on Ethernet controller

Use the following sample configuration to display PRBS details configured on an Ethernet controller:

```
RP/0/RP0/CPU0:ios#show controllers fourHundredGigEctrlr 0/0/0/4
Fri Sep 25 09:45:18.222 UTC
Operational data for interface FourHundredGigEctrlr0/0/0/4:

State:
  Administrative state: enabled
  Operational state: Up
  LED state: Green On
  Maintenance: Enabled
  AINS Soak: None
    Total Duration: 0 hour(s) 0 minute(s)
    Remaining Duration: 0 hour(s) 0 minute(s) 0 second(s)
  PRBS:
    Status: Locked
    Mode: Source-sink
    Pattern: PN31
    Direction: Line
    Framing: Framed
    Configured Time: 25 Sep 09:44:56 (22 seconds elapsed)
    First Lock Established Time: 25 Sep 09:44:59 (19 seconds elapsed)
    Lock Time(in seconds): 19
    Bit errors: 0
    Lock Found count: 1
    Lock Lost count: 0
    Result: PASS
  Laser Squelch: Disabled
  Insert Idle Ingress: Disabled
  Insert Idle Egress: Disabled
  State transition count: 1
  Last link flapped: 00:29:47
```

### Example to view PRBS details on CoherentDSP controller

Use the following sample configuration to display PRBS details configured on a coherentDSP controller:

```
RP/0/RP0/CPU0:ios#show controllers coherentDSP 0/0/0/7 prbs-details
Wed Nov 15 18:13:35.210 UTC
```

```
-----PRBS details-----
PRBS Test           : Enable
PRBS Mode           : Source-Sink
PRBS Pattern        : PN15
PRBS Status         : Locked
```

### Example to view PRBS details on ODU controller

Use the following sample configuration to display PRBS details configured on an ODU controller where the PRBS status is displayed as **Not Applicable**, when the mode is **Source**.

## Viewing PRBS Performance Monitoring Parameters

```
RP/0/RP0/CPU0:ios#show controllers ODU4 0/3/0/8 prbs-details
Mon Jan 11 05:29:12.436 UTC
```

```
-----PRBS details-----
PRBS Test           : Enable
PRBS Mode           : Source
PRBS Pattern        : PN7
PRBS Status         : Not Applicable
-----
```

## Viewing PRBS Performance Monitoring Parameters

To view the PRBS performance monitoring parameters on a coherentDSP, ODU, or Ethernet controller, use the following command:

**show controllers coherentDSP | ODU4 R/S/I/P pm {current | history} {15-min|24-hour} prbs**

Following is an example of how to view the cumulative count of PRBS bit errors in the 15-min sampling interval on a CoherentDSP controller:

```
RP/0/RP0/CPU0:ios#show controllers coherentDSP 0/0/0/7 pm current 15-min prbs
Wed Nov 15 18:19:10.308 UTC
```

```
PRBS in the current interval [18:15:00 - 18:19:10 Wed Nov 15 2023]
```

```
PRBS current bucket type : Valid
```

```
EBC           : 0           Threshold : 0           TCA(enable)  : NO
FOUND-COUNT   : 0           Threshold : 0           TCA(enable)  : NO
LOST-COUNT    : 0           Threshold : 0           TCA(enable)  : NO
```

```
FOUND-AT-TS   : NULL
LOST-AT-TS    : NULL
```

```
CONFIG-PTRN   : PRBS_PATTERN_PN15
STATUS        : LOCKED
```

```
Last clearing of "show controllers OTU" counters never
```

Following is an example of how to view PRBS performance monitoring parameters for a 15-minute sampling interval on an ODU controller:

```
RP/0/RP0:ios#show controllers ODU4 0/3/0/1 pm current 15-min prbs
Mon Jan 11 00:58:48.327 UTC
```

```
PRBS in the current interval [00:45:00 - 00:58:48 Mon Jan 11 2021]
```

```
PRBS current bucket type : Valid
```

```
EBC           : 40437528165
FOUND-COUNT   : 1 FOUND-AT-TS : 00:51:22 Mon Jan 11 2021
LOST-COUNT    : 1 LOST-AT-TS  : 00:52:52 Mon Jan 11 2021
CONFIG-PTRN   : PRBS_PATTERN_PN7
```

```
Last clearing of "show controllers ODU" counters never
```

The following tables describes the fields of PRBS PM parameters.

Following is an example of how to view PRBS performance monitoring parameters for a 15-minute sampling interval on an Ethernet controller:

```
RP/0/RP0/CPU0:ios#show controllers fourHundredGigEctr1r 0/0/0/4 pm current 15-min prbs
Wed Nov 15 18:48:19.114 UTC
```

```
PRBS in the current interval [18:45:00 - 18:48:19 Wed Nov 15 2023]
```

```
PRBS current bucket type : Valid
```

```
EBC           : 0           Threshold : 0           TCA(enable)  : NO
FOUND-COUNT   : 0           Threshold : 0           TCA(enable)  : NO
LOST-COUNT    : 0           Threshold : 0           TCA(enable)  : NO
```

```
FOUND-AT-TS   : NULL
LOST-AT-TS    : NULL
```

```
CONFIG-PTRN   : PRBS_PATTERN_PN23
STATUS        : LOCKED
```

Last clearing of "show controllers ETHERNET" counters never

The following tables describes the fields of PRBS PM parameters.

**Table 1: PRBS PM Parameters**

PM Parameter	Description
EBC	Cumulative count of PRBS bit errors in the sampling interval (15-minute or 24-hour). PRBS bit errors are accumulated only if PRBS signal is locked.
FOUND-COUNT	Number of state transitions from signal unlocked state to signal locked state in the sampling interval. If state change is not observed in the interval, the count is 0.
LOST-COUNT	Number of state transitions from signal locked state to signal unlocked state in the sampling interval. If state change is not observed in the interval, the count is 0.
FOUND-AT-TS	Latest timestamp when the PRBS state moves from unlocked state to locked state in the sampling interval. If state change is not observed in the interval, the value is null.
CONFIG-PTRN	Configured PRBS pattern on the port.
STATUS	Displays the PRBS status.

## Clearing Bit Errors and Lock Time for PRBS

Lock time is the time that is elapsed since the last PRBS lock is detected.

The following sample shows that bit errors are observed during the PRBS test:

```
RP/0/RP0/CPU0:ios#show controllers odu4 0/2/0/5 prbs-details
Fri Nov 13 03:21:44.191 UTC
```

```

-----PRBS details-----
PRBS Test : Enable
PRBS Mode : Source-Sink
PRBS Pattern : INVERTED PN31
PRBS Status : Locked
PRBS Direction : Line
PRBS Lock Time(in seconds) : 28
PRBS Bit Errors : 23776

```

To clear the lock time and bit errors before the PRBS test, use the **clear** command:

```

RP/0/RP0/CPU0:ios#clear controller odu4 0/2/0/5 prbs-details
Fri Nov 13 03:21:50.726 UTC
PRBS bit errors cleared

```

The following sample displays the bit errors and lock time are removed.

```

RP/0/RP0/CPU0:ios#show controllers odu4 0/2/0/5 prbs-details
Fri Nov 14 03:21:44.191 UTC
-----PRBS details-----
PRBS Test : Enable
PRBS Mode : Source-Sink
PRBS Pattern : INVERTED PN31
PRBS Status : Locked
PRBS Direction : Line
PRBS Lock Time(in seconds) : 2
PRBS Bit Errors : 0

```

## Trunk PRBS

NCS 1014 trunk port supports the following PRBS patterns:

- **PRBS31** — Sequence length is from  $2^{31} - 1$  bits.
- **PRBS23** — Sequence length is from  $2^{23} - 1$  bits.
- **PRBS15** — Sequence length is from  $2^{15} - 1$  bits.
- **PRBS11** — Sequence length is from  $2^{11} - 1$  bits.
- **PRBS7** — Sequence length is from  $2^7 - 1$  bits.



**Tip** We recommend that for higher datarates like 100G and 400G:

- use high sequence length PRBS patterns and
- use PRBS inverted pattern.



**Note** Interoperability for ethernet PRBS PN23 pattern is not supported in the 2.4T and 2.4TX cards.



**Note** In case of muxponder configuration, PRBS is not supported on the split ports of the 2.4TX card.

### Configuring Trunk PRBS on NCS1K4-2.4T-K9

Use the following sample configuration to configure PRBS trunk mode on the NCS1K4-2.4T-K9:

```
RP/0/RP0/CPU0:ios(config)#controller CoherentDSP0/0/0/7
RP/0/RP0/CPU0:ios(config-CoDSP)#secondary-admin-state maintenance
RP/0/RP0/CPU0:ios(config-CoDSP)#prbs mode source-sink pattern pn15
RP/0/RP0/CPU0:ios(config-CoDSP)#commit
Wed Nov 15 18:11:55.450 UTC
```

**Table 2: Feature History**

Feature Name	Release Information	Description
Cumulative PRBS on CoherentDSP controllers	Cisco IOS XR Release 24.3.1	The cumulative PRBS (Pseudo-Random Binary Sequence) on CoherentDSP controllers enhances troubleshooting capabilities between the trunk ports.  Show coherentDSP R/S/I/P prbs-details command output now includes the newly supported fields.

Use the following sample configuration to display PRBS details:

```
RP/0/RP0/CPU0:ios#show controllers coherentDSP 0/0/0/7 prbs-details
Wed Nov 15 18:13:35.210 UTC
```

```
-----PRBS details-----
PRBS Test           : Enable
PRBS Mode           : Source-Sink
PRBS Pattern        : PN15
PRBS Status         : Locked
PRBS Lock Time(in seconds) : 37
PRBS Bit Errors     : 0
PRBS Found Count    : 1
PRBS Lost Count     : 0
PRBS Configured Time : 11 Feb 00:20:43 (719 seconds elapsed)
PRBS First Lock Established Time: 11 Feb 00:32:05 (37 seconds elapsed)
Result Summary      : PASS
```

The **Result Summary** will display **PASS** if the PRBS bit errors are 0 and the PRBS elapsed lock time is equal to the elapsed first lock established time.

Use this command to clear the counters .

```
clear controller coherentDSP 0/0/0/7 prbs-details
```

Use the following sample configuration to display cumulative count of PRBS bit errors in the 15-min sampling interval:

```
RP/0/RP0/CPU0:ios#show controllers coherentDSP 0/0/0/7 pm current 15-min prbs
Wed Nov 15 18:19:10.308 UTC
```

```
PRBS in the current interval [18:15:00 - 18:19:10 Wed Nov 15 2023]
```

```

PRBS current bucket type : Valid

EBC           : 0           Threshold : 0           TCA(enable)  : NO
FOUND-COUNT   : 0           Threshold : 0           TCA(enable)  : NO
LOST-COUNT    : 0           Threshold : 0           TCA(enable)  : NO

FOUND-AT-TS   : NULL
LOST-AT-TS    : NULL

CONFIG-PTRN   : PRBS_PATTERN_PN15
STATUS        : LOCKED

Last clearing of "show controllers OTU" counters never

```

## Client PRBS

NCS 1014 client port supports the following PRBS patterns:

- **PRBS31** — Sequence length is from  $2^{31} - 1$  bits.
- **PRBS23** — Sequence length is from  $2^{23} - 1$  bits.

**Table 3: Feature History**

Feature Name	Release Information	Description
New PRBS Counters on Ethernet Controllers	Cisco IOS XR Release 24.4.1	<p>The new Pseudo-Random Binary Sequence (PRBS) counters on Ethernet controllers collect statistics in a cumulative manner.</p> <p>The output of the <code>show controllers controller-type R/S/I/P</code> command now includes these counters:</p> <ul style="list-style-type: none"> <li>• Configured Time</li> <li>• First Lock Established Time.</li> <li>• Lock Time (in seconds)</li> <li>• Bit Errors</li> <li>• Lock Found Count</li> <li>• Lock the Lost Count</li> <li>• Result</li> </ul>

### Configuring Client PRBS on NCS1K4-2.4T-K9

Use the following sample configuration to configure PRBS client mode on the NCS1K4-2.4T-K9:

```

RP/0/RP0/CPU0:ios(config)#controller fourHundredGigECtrlr 0/2/0/4
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#prbs mode source-sink pattern <pn31|pn23> direction
system <line|system>

```



```
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#sec-admin-state maintenance
RP/0/RP0/CPU0:ios(config-eth-ctrlr)#commit
```

Use the following sample configuration to display four hundred gigabit client controllers details:

```
RP/0/RP0/CPU0:ios#show controllers fourHundredGigEctrlr 0/0/0/4
Fri Sep 25 09:48:03.959 UTC
Operational data for interface FourHundredGigEctrlr0/0/0/4:

State:
  Administrative state: enabled
  Operational state: Up
  LED state: Green On
  Maintenance: Enabled
  AINS Soak: None
    Total Duration: 0 hour(s) 0 minute(s)
    Remaining Duration: 0 hour(s) 0 minute(s) 0 second(s)
  PRBS:
    Status: Locked
    Mode: Source-sink
    Pattern: PN31
    Direction: Line
    Framing: Framed
    Configured Time: 25 Sep 09:44:56 (187 seconds elapsed)
    First Lock Established Time: 25 Sep 09:47:52 (11 seconds elapsed)
    Lock Time(in seconds): 11
    Bit errors: 0
    Lock Found count: 0
    Lock Lost count: 0
    Result: PASS
  Laser Squelch: Disabled
  Insert Idle Ingress: Disabled
  Insert Idle Egress: Disabled
  State transition count: 1
  Last link flapped: 00:32:32
```

The **Result** displays **PASS**, only if the **First Lock Established Time** and **Lock Time (in seconds)** are same, and **Bit errors** is 0.

Use the following sample configuration to clear four hundred gigabit client controller PRBS statistics:

```
RP/0/RP0/CPU0:ios#clear controller fourHundredGigEctrlr 0/0/0/4 prbs-stats
Fri Sep 25 09:47:52.678 UTC
Operational data for interface FourHundredGigEctrlr0/0/0/4:
State:
  Administrative state: enabled
  Operational state: Up
  LED state: Green On
  Maintenance: Enabled
  AINS Soak: None
    Total Duration: 0 hour(s) 0 minute(s)
    Remaining Duration: 0 hour(s) 0 minute(s) 0 second(s)
  PRBS:
    Status: Locked
    Mode: Source-sink
    Pattern: PN31
    Direction: Line
    Framing: Framed
    Configured Time: 25 Sep 09:44:56 (179 seconds elapsed)
    First Lock Established Time: 25 Sep 09:47:52 (3 seconds elapsed)
    Lock Time(in seconds): 3
    Bit errors: 0
    Lock Found count: 0
    Lock Lost count: 0
```

```
Result: PASS
Laser Squelch: Disabled
Insert Idle Ingress: Disabled
Insert Idle Egress: Disabled
State transition count: 1
Last link flapped: 00:32:24
```

Use the following sample configuration to display four hundred gigabit client controller PRBS bit errors in the 15-min sampling interval:

```
RP/0/RP0/CPU0:ios#show controllers fourHundredGigEctrlr 0/0/0/4 pm current 15-min prbs
Wed Nov 15 18:48:19.114 UTC
PRBS in the current interval [18:45:00 - 18:48:19 Wed Nov 15 2023]
PRBS current bucket type : Valid
EBC          : 0          Threshold : 0          TCA(enable) : NO
FOUND-COUNT  : 0          Threshold : 0          TCA(enable) : NO
LOST-COUNT   : 0          Threshold : 0          TCA(enable) : NO
FOUND-AT-TS  : NULL
LOST-AT-TS   : NULL
CONFIG-PTRN  : PRBS_PATTERN_PN23
STATUS       : LOCKED
```

Last clearing of "show controllers ETHERNET" counters never