# Configuring the Card Mode

This chapter lists the supported configurations and the procedures to configure the card mode on the line cards.

**Note** Unless otherwise specified, "line cards" refers to 1.2T and 1.2TL line cards.

# 1.2T Line Card

The following section describes the supported configurations and procedures to configure the card modes on the line card.

## Card Modes

The line cards support module and slice configurations.

The line cards have two trunk ports (0 and 1) and 12 client ports (2 through 13) each. You can configure the line card in two modes:

- Muxponder—In this mode, both trunk ports are configured with the same trunk rate. The client-to-trunk mapping is in a sequence.
- Muxponder slice—In this mode, each trunk port is configured independent of the other with different trunk rates. The client-to-trunk mapping is fixed. For Trunk 0, the client ports are 2 through 7. For Trunk 1, the client ports are 8 through 13.

## Sub 50G Configuration

You can configure the sub 50G or coupled mode on the line card only in the muxponder mode. The following table displays the port configuration for the supported data rates.

| Trunk Data Rate (per trunk) | Total Configured Data rate | Card Support | Trunk Ports | Client Ports for Trunk 0 (100G) | Shared Client Port (50G per trunk) | Client Ports for Trunk 1 (100G) |
|---|---|---|---|---|---|---|
| 50G | 100G | 1.2T | 0, 1 | - | 2 | - |
| 150G | 300G | 1.2T | 0, 1 | 2 | 3 | 4 |
| 350G | 700G | 1.2T | 0, 1 | 2, 3, 4 | 5 | 6, 7, 8 |
| 450G | 900G | 1.2T | 0, 1 | 2, 3, 4, 5 | 6 | 7, 8, 9, 10 |
| 550G | 1.1T | 1.2T | 0, 1 | 2, 3, 4, 5, 6 | 7 | 8, 9, 10, 11, 12 |

1.2T line card supports an alternate port configuration for Sub 50G (split client port mapping) that you configure using CLI. The following table displays the port configuration for the supported data rates.

| Trunk Data Rate (per trunk) | Total Configured Data rate | Card Support | Trunk Ports | Client Ports for Trunk 0 (100G) | Shared Client Port (50G per trunk) | Client Ports for Trunk 1 (100G) |
|---|---|---|---|---|---|---|
| 50G | 100G | 1.2T | 0, 1 | - | 7 | - |
| 150G | 300G | 1.2T | 0, 1 | 2 | 7 | 8 |
| 250G | 500G | 1.2T | 0, 1 | 2, 3 | 7 | 8, 9 |
| 350G | 700G | 1.2T | 0, 1 | 2, 3, 4 | 7 | 8, 9, 10 |
| 450G | 900G | 1.2T | 0, 1 | 2, 3, 4, 5 | 7 | 8, 9, 10, 11 |
| 550G | 1.1T | 1.2T | 0, 1 | 2, 3, 4, 5, 6 | 7 | 8, 9, 10, 11, 12 |

**Note** In all x50G configurations, client traffic on the middle port is affected with ODUK-BDI and LF alarms after the **power cycle or link flap** on the trunk side. This issue is raised when the two network lanes work in coupled mode and move from low to high power. To solve this issue, create a new frame either at the near-end or far-end by performing **shut** or **no shut** of the trunk ports.

### Coupled Mode Restrictions

The following restrictions apply to the coupled mode configuration:

- Both trunk ports must be configured with the same bits-per-symbol or baud rate and must be sent over same fiber and direction.
- The chromatic dispersion must be configured to the same value for both trunk ports.
- When trunk internal loopback is configured, it must be done for both trunk ports. Configuring internal loopback on only one trunk results in traffic loss.

- Fault on a trunk port of a coupled pair may cause errors on all clients including those running only on the unaffected trunk port.

## Configure Split Client Port Mapping

You can configure the trunk port to client port mapping for sub 50G data ratesin the default mode or in the split client port mapping mode.

To configure the split client port mapping, use the following commands.

**configure**

**hw-module location** *location* **mxponder**

**split-client-port-mapping**

**commit**

The following is a sample in which split-client-port-mapping is configured with a 450G trunk payload.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#split-client-port-mapping
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#end
```

To remove the split client port-mapping configuration and configure default client port mapping, use the following commands.

**configure**

**hw-module location** *location* **mxponder**

**no split-client-port-mapping**

**commit**

The following is a sample in which split client port-mapping configuration is removed.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#no split-client-port-mapping
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#end
```

### Verifying the Port Mapping Configuration

The following is a sample ouput of the split client port-mapping.

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder

Location:             0/1/NXR0
Client Bitrate:       100GE
Trunk  Bitrate:       450G
Status:               Provisioning In Progress
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/0
CoherentDSP0/1/0/1
                                Traffic Split Percentage

HundredGigECtrlr0/1/0/2         ODU40/1/0/0/1                        100
       0
HundredGigECtrlr0/1/0/3         ODU40/1/0/0/2                        100
```

```
       0
HundredGigECtrlr0/1/0/4         ODU40/1/0/0/3                        100
       0
HundredGigECtrlr0/1/0/5         ODU40/1/0/0/4                        100
       0
HundredGigECtrlr0/1/0/7         ODU40/1/0/0/5                         50
      50
HundredGigECtrlr0/1/0/8         ODU40/1/0/1/1                          0
     100
HundredGigECtrlr0/1/0/9         ODU40/1/0/1/2                          0
     100
HundredGigECtrlr0/1/0/10        ODU40/1/0/1/3                          0
     100
HundredGigECtrlr0/1/0/11        ODU40/1/0/1/4                          0
     100
```

The following is a sample ouput of the default client port mapping.

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder

Location:             0/1/NXR0
Client Bitrate:       100GE
Trunk  Bitrate:       450G
Status:               Provisioning In Progress
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/0
CoherentDSP0/1/0/1
                                Traffic Split Percentage

HundredGigECtrlr0/1/0/2         ODU40/1/0/0/1                        100
       0
HundredGigECtrlr0/1/0/3         ODU40/1/0/0/2                        100
       0
HundredGigECtrlr0/1/0/4         ODU40/1/0/0/3                        100
       0
HundredGigECtrlr0/1/0/5         ODU40/1/0/0/4                        100
       0
HundredGigECtrlr0/1/0/6         ODU40/1/0/0/5                         50
      50
HundredGigECtrlr0/1/0/7         ODU40/1/0/1/1                          0
     100
HundredGigECtrlr0/1/0/8         ODU40/1/0/1/2                          0
     100
HundredGigECtrlr0/1/0/9         ODU40/1/0/1/3                          0
     100
HundredGigECtrlr0/1/0/10        ODU40/1/0/1/4                          0
     100
```

# Supported Data Rates

The following data rates are supported on the line card.

The following table displays the client and trunk ports that are enabled for the muxponder configuration.

| Trunk Data Rate | Card Support | Client Data Rate (100GE, OTU4) | Trunk Ports | Client Ports |
|---|---|---|---|---|
| 100 | 1.2T | 100GE, OTU4 | 0 | 2 |
| 200 | 1.2T | 100GE, OTU4 | 0, 1 | 2, 3, 4, 5 |

| Trunk Data Rate | Card Support | Client Data Rate (100GE, OTU4) | Trunk Ports | Client Ports |
|---|---|---|---|---|
| 300 | 1.2T | 100GE, OTU4 | 0, 1 | 2, 3, 4, 5, 6, 7 |
| 400 | 1.2T | 100GE, OTU4 | 0, 1 | 2, 3, 4, 5, 6, 7, 8, 9 |
| 500 | 1.2T | 100GE, OTU4 | 0, 1 | 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 |
| 600 | 1.2T | 100GE, OTU4 | 0, 1 | 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 |

The following table displays the client and trunk ports that are enabled for the muxponder slice 0 configuration.

| Trunk Data Rate | Card Support | Client Data Rate | Trunk Ports | Client Ports |
|---|---|---|---|---|
| 100 | 1.2T | 100, OTU4 | 0 | 2 |
| 200 | 1.2T | 100, OTU4 | 0 | 2, 3 |
| 300 | 1.2T | 100, OTU4 | 0 | 2, 3, 4 |
| 400 | 1.2T | 100, OTU4 | 0 | 2, 3, 4, 5 |
| 500 | 1.2T | 100, OTU4 | 0 | 2, 3, 4, 5, 6 |
| 600 | 1.2T | 100, OTU4 | 0 | 2, 3, 4, 5, 6, 7 |

The following table displays the client and trunk ports that are enabled for the muxponder slice 1 configuration.

| Trunk Data Rate | Card Support | Client Data Rate | Trunk Ports | Client Ports |
|---|---|---|---|---|
| 100 | 1.2T | 100, OTU4 | 1 | 8 |
| 200 | 1.2T | 100, OTU4 | 1 | 8, 9 |
| 300 | 1.2T | 100, OTU4 | 1 | 8, 9, 10 |
| 400 | 1.2T | 100, OTU4 | 1 | 8, 9, 10, 11 |
| 500 | 1.2T | 100, OTU4 | 1 | 8, 9, 10, 11, 12 |
| 600 | 1.2T | 100, OTU4 | 1 | 8, 9, 10, 11, 12, 13 |

All configurations can be accomplished by using appropriate values for client bitrate and trunk bitrate parameters of the **hw-module** command.

The following table displays the trunk parameter ranges for the 1.2T card.

| Trunk Payload | FEC | Min BPS | Max BPS | Min GBd | Max GBd |
|---|---|---|---|---|---|
| 50G | 15% | 1 | 1.3125 | 24.0207911 | 31.5272884 |
| 50G | 27% | 1 | 1.4453125 | 24.0207911 | 34.7175497 |

| Trunk Payload | FEC | Min BPS | Max BPS | Min GBd | Max GBd |
|---|---|---|---|---|---|
| 100G | 15% | 1 | 2.625 | 24.0207911 | 63.0545768 |
| 100G | 27% | 1 | 2.890625 | 24.0207911 | 69.4350994 |
| 150G | 15% | 1.3203125 | 3.9375 | 24.0207911 | 71.6359689 |
| 150G | 27% | 1.453125 | 4.3359375 | 24.0207911 | 71.6749413 |
| 200G | 15% | 1.7578125 | 5.25 | 24.0207911 | 71.7420962 |
| 200G | 27% | 2 | 4.40625 | 31.51 | 69.43 |
| 250G | 15% | 2.1953125 | 6 | 26.2727403 | 71.8059237 |
| 250G | 27% | 2.4140625 | 6 | 28.9312914 | 71.9068991 |
| 300G | 15% | 2.6328125 | 6 | 31.5272884 | 71.8485385 |
| 300G | 27% | 2.8984375 | 6 | 34.7175497 | 71.8681352 |
| 350G | 15% | 3.0703125 | 6 | 36.7818364 | 71.8790086 |
| 350G | 27% | 3.3828125 | 6 | 40.503808 | 71.8404724 |
| 400G | 15% | 3.5078125 | 6 | 42.0363845 | 71.9018782 |
| 400G | 27% | 3.8671875 | 6 | 46.2900663 | 71.8197392 |
| 450G | 15% | 3.9453125 | 6 | 47.2909326 | 71.9196757 |
| 450G | 27% | 4.34375 | 6 | 52.0763245 | 71.9327648 |
| 500G | 15% | 4.3828125 | 6 | 52.5454806 | 71.93392 |
| 500G | 27% | 4.8281250 | 6 | 57.8625828 | 71.9068991 |
| 550G | 15% | 4.8203125 | 6 | 57.8000287 | 71.9455787 |
| 550G | 27% | 5.3125 | 6 | 63.6488411 | 71.88575 |
| 600G | 15% | 5.2578125 | - | - | 71.9552971 |

| Trunk Payload | FEC | Min BPS | Max BPS | Min GBd | Max GBd |
|---|---|---|---|---|---|
| 100G | 15% | 1 | 2.625 | 24.0207911 | 63.0545768 |
| 100G | 27% | 1 | 2.890625 | 24.0207911 | 69.4350994 |
| 150G | 15% | 1.3203125 | 3.9375 | 24.0207911 | 71.6359689 |
| 150G | 27% | 1.453125 | 4.3359375 | 24.0207911 | 71.6749413 |
| 200G | 15% | 2 | 4 | 31.5272884 | 63.0545768 |

| Trunk Payload | FEC | Min BPS | Max BPS | Min GBd | Max GBd |
|---|---|---|---|---|---|
| 200G | 27% | 2 | 4.40625 | 31.51664088 | 69.43509943 |
| 250G | 15% | 2.1953125 | 4.5 | 35.0303204 | 71.8059237 |
| 250G | 27% | 2.4140625 | 4.5 | 38.5750552 | 71.9068991 |
| 300G | 15% | 2.6328125 | 4.5 | 42.0363845 | 71.8485385 |
| 300G | 27% | 2.8984375□ | 4.5 | 46.2900662857142 | 71.86813526 |
| 350G | 15% | 3.0703125 | 4.5 | 49.0424486 | 71.8790086 |
| 350G | 27% | 3.3828125 | 4.5 | 54.0050773 | 71.8404724 |
| 400G | 15% | 3.5078125 | 4.5 | 56.0485127 | 71.9018782 |
| 400G | 27% | 3.8671875 | 4.5 | 61.72008838 | 71.81973921 |

# Configuring the Card Mode

You can configure the line card in the module (muxponder) or slice configuration (muxponder slice).

To configure the card in the muxponder mode, use the following commands.

**configure**

**hw-module location** *location* **mxponder client-rate {100GE | OTU4}**

**hw-module location** *location* **mxponder trunk-rate** {**50G | 100G150G | 200G | 250G | 300G | 350G | 400G | 450G | 500G | 550G | 600G** }

**commit**

To configure the card in the muxponder slice mode, use the following commands. **configure hw-module location** *location* **mxponder-slice** *mxponder-slice-number* **client-rate { 100GE|OTU4}**

**hw-module location** *location* **mxponder-slice trunk-rate** { **100G | 200G | 300G | 400G | 500G | 600G** }

**commit**

### Examples

The following is a sample in which the card is configured in the muxponder mode with a 550G trunk payload.

```
RP/0/RP0/CPU0:ios#config
Tue Oct 15 01:24:56.355 UTC
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder client-rate 100GE
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder trunk-rate 550G
RP/0/RP0/CPU0:ios(config)#commit
```

The following is a sample in which the card is configured in the muxponder mode with a 500G trunk payload.

```
RP/0/RP0/CPU0:ios#config
Sun Feb 24 14:09:33.989 UTC
RP/0/RP0/CPU0:ios(config)#hw-module location 0/2/NXR0 mxponder client-rate OTU4
```

```
RP/0/RP0/CPU0:ios(config)#hw-module location 0/2/NXR0 mxponder trunk-rate 500G
RP/0/RP0/CPU0:ios(config)#commit
```

The following is a sample in which the card is configured in the muxponder slice 0 mode with a 500G trunk payload.

```
RP/0/RP0/CPU0:ios#config
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 0 client-rate 100GE
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 0 trunk-rate 500G
RP/0/RP0/CPU0:ios(config)#commit
```

The following is a sample in which the card is configured in the muxponder slice 1 mode with a 400G trunk payload.

```
RP/0/RP0/CPU0:ios#config
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 1 client-rate 100GE
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 1 trunk-rate 400G
RP/0/RP0/CPU0:ios(config)#commit
```

The following is a sample in which the card is configured with mixed client rates in the muxponder slice mode.

```
RP/0/RP0/CPU0:ios#configure
Mon Mar 23 06:10:22.227 UTC
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 0 client-rate OTU4
trunk-rate 500G
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 1 client-rate 100GE
trunk-rate 500G
RP/0/RP0/CPU0:ios(config)#commit
```

### Verifying the Card Configuration

```
RP/0/RP0/CPU0:ios#show hw-module location 0/2/NXR0 mxponder
Fri Mar 15 11:48:48.344 IST

Location:             0/2/NXR0
Client Bitrate:       100GE
Trunk  Bitrate:       500G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
Client Port             Mapper/Trunk Port   CoherentDSP0/2/0/0  CoherentDSP0/2/0/1
                     Traffic Split Percentage

HundredGigECtrlr0/2/0/2  ODU40/2/0/0/1              100                  0
HundredGigECtrlr0/2/0/3  ODU40/2/0/0/2              100                  0
HundredGigECtrlr0/2/0/4  ODU40/2/0/0/3              100                  0
HundredGigECtrlr0/2/0/5  ODU40/2/0/0/4              100                  0
HundredGigECtrlr0/2/0/6  ODU40/2/0/0/5              100                  0
HundredGigECtrlr0/2/0/7  ODU40/2/0/1/1                0                100
HundredGigECtrlr0/2/0/8  ODU40/2/0/1/2                0                100
HundredGigECtrlr0/2/0/9  ODU40/2/0/1/3                0                100
HundredGigECtrlr0/2/0/10 ODU40/2/0/1/4                0                100
HundredGigECtrlr0/2/0/11 ODU40/2/0/1/5                0                100
```

The following is a sample ouput of the coupled mode configuration where the shared client port is highlighted.

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder
Tue Oct 15 01:25:57.358 UTC

Location:             0/1/NXR0
Client Bitrate:       100GE
Trunk  Bitrate:       550G
```

```
Status:               Provisioned
LLDP Drop Enabled:    FALSE
Client Port           Mapper/Trunk Port    CoherentDSP0/1/0/0 CoherentDSP0/1/0/1
                   Traffic Split Percentage

HundredGigECtrlr0/1/0/2    ODU40/1/0/0/1             100                  0
HundredGigECtrlr0/1/0/3    ODU40/1/0/0/2             100                  0
HundredGigECtrlr0/1/0/4    ODU40/1/0/0/3             100                  0
HundredGigECtrlr0/1/0/5    ODU40/1/0/0/4             100                  0
HundredGigECtrlr0/1/0/6    ODU40/1/0/0/5             100                  0
HundredGigECtrlr0/1/0/7    ODU40/1/0/0/6              50                 50
HundredGigECtrlr0/1/0/8    ODU40/1/0/1/1               0                100
HundredGigECtrlr0/1/0/9    ODU40/1/0/1/2               0                100
HundredGigECtrlr0/1/0/10   ODU40/1/0/1/3               0                100
HundredGigECtrlr0/1/0/11   ODU40/1/0/1/4               0                100
HundredGigECtrlr0/1/0/12   ODU40/1/0/1/5               0                100
```

The following is a sample ouput of all the muxponder slice 0 configurations.

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice  0
Fri Mar 15 06:04:18.348 UTC

Location:             0/1/NXR0
Slice ID:             0
Client Bitrate:       100GE
Trunk  Bitrate:       500G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/0
                                Traffic Split Percentage

HundredGigECtrlr0/1/0/2         ODU40/1/0/0/1                      100
HundredGigECtrlr0/1/0/3         ODU40/1/0/0/2                      100
HundredGigECtrlr0/1/0/4         ODU40/1/0/0/3                      100
HundredGigECtrlr0/1/0/5         ODU40/1/0/0/4                      100
HundredGigECtrlr0/1/0/6         ODU40/1/0/0/5                      100
```

The following is a sample ouput of all the muxponder slice 1 configurations.

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 1
Fri Mar 15 06:11:50.020 UTC

Location:             0/1/NXR0
Slice ID:             1
Client Bitrate:       100GE
Trunk  Bitrate:       400G
Status:               Provisioned
LLDP Drop Enabled:    TRUE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/1
                                Traffic Split Percentage

HundredGigECtrlr0/1/0/8         ODU40/1/0/1/1                      100
HundredGigECtrlr0/1/0/9         ODU40/1/0/1/2                      100
HundredGigECtrlr0/1/0/10        ODU40/1/0/1/3                      100
HundredGigECtrlr0/1/0/11        ODU40/1/0/1/4                      100
```

The following is a sample ouput of the muxponder slice 1 configuration with client configured as OTU4.

```
RP/0/RP0/CPU0:ios#sh hw-module location 0/0/NXR0 mxponder-slice 1

Wed Mar 11 13:59:11.073 UTC

Location:             0/0/NXR0
Slice ID:             1
```

```
Client Bitrate:       OTU4
Trunk  Bitrate:       200G
Status:               Provisioned
Client Port                     Peer/Trunk Port            CoherentDSP0/0/0/1
                              Traffic Split Percentage
OTU40/0/0/8                     ODU40/0/0/1/1                      100
OTU40/0/0/9                     ODU40/0/0/1/2                      100
```

The following is a sample to verify the mixed client rate configuration in the muxponder slice mode.

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder
Mon Mar 23 06:20:22.227 UTC

Location:             0/1/NXR0
Slice ID:             0
Client Bitrate:       OTU4
Trunk  Bitrate:       500G
Status:               Provisioned
Client Port                     Peer/Trunk Port            CoherentDSP0/1/0/0
                                Traffic Split Percentage

OTU40/1/0/2                     ODU40/1/0/0/1                      100
OTU40/1/0/3                     ODU40/1/0/0/2                      100
OTU40/1/0/4                     ODU40/1/0/0/3                      100
OTU40/1/0/5                     ODU40/1/0/0/4                      100
OTU40/1/0/6                     ODU40/1/0/0/5                      100


Location:             0/1/NXR0
Slice ID:             1
Client Bitrate:       100GE
Trunk  Bitrate:       500G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/1
                                Traffic Split Percentage

HundredGigECtrlr0/1/0/8         ODU40/1/0/1/1                         100
HundredGigECtrlr0/1/0/9         ODU40/1/0/1/2                         100
HundredGigECtrlr0/1/0/10        ODU40/1/0/1/3                         100
HundredGigECtrlr0/1/0/11        ODU40/1/0/1/4                         100
HundredGigECtrlr0/1/0/12        ODU40/1/0/1/5                         100
```

Use the following command to clear alarm statistics on the optics or coherent DSP controller.

**clear counters controller** *controllertype R/S/I/P*

The following is a sample in which the alarm statistics are cleared on the coherent DSP controller.

```
RP/0/RP0/CPU0:ios#show controller coherentDSP 0/1/0/0
Tue Jun 11 05:15:12.540 UTC

Port                                            : CoherentDSP 0/1/0/0
Controller State                                : Up
Inherited Secondary State                       : Normal
Configured Secondary State                      : Normal
Derived State                                   : In Service
Loopback mode                                   : None
BER Thresholds                                  : SF = 1.0E-5  SD = 1.0E-7
Performance Monitoring                          : Enable

Alarm Information:
```

```
LOS = 1 LOF = 1 LOM = 0
OOF = 1 OOM = 1 AIS = 0
IAE = 0 BIAE = 0        SF_BER = 0
SD_BER = 2      BDI = 2 TIM = 0
FECMISMATCH = 0 FEC-UNC = 0
Detected Alarms                                 : None

Bit Error Rate Information
PREFEC  BER                                     : 8.8E-03
POSTFEC BER                                     : 0.0E+00

TTI :
        Remote hostname                         : P2B8
        Remote interface                        : CoherentDSP 0/1/0/0
        Remote IP addr                          : 0.0.0.0

FEC mode                                        : Soft-Decision 15

AINS Soak                                       : None
AINS Timer                                      : 0h, 0m
AINS remaining time                             : 0 seconds
RP/0/RP0/CPU0:ios#clear counters controller coherentDSP 0/1/0/0
Tue Jun 11 05:17:07.271 UTC
All counters are cleared
RP/0/RP0/CPU0:ios#show controllers coherentDSP 0/1/0/1
Tue Jun 11 05:20:55.199 UTC

Port                                            : CoherentDSP 0/1/0/1
Controller State                                : Up
Inherited Secondary State                       : Normal
Configured Secondary State                      : Normal
Derived State                                   : In Service
Loopback mode                                   : None
BER Thresholds                                  : SF = 1.0E-5  SD = 1.0E-7
Performance Monitoring                          : Enable

Alarm Information:
LOS = 0 LOF = 0 LOM = 0
OOF = 0 OOM = 0 AIS = 0
IAE = 0 BIAE = 0        SF_BER = 0
SD_BER = 0      BDI = 0 TIM = 0
FECMISMATCH = 0 FEC-UNC = 0
Detected Alarms                                 : None

Bit Error Rate Information
PREFEC  BER                                     : 1.2E-02
POSTFEC BER                                     : 0.0E+00

TTI :
        Remote hostname                         : P2B8
        Remote interface                        : CoherentDSP 0/1/0/1
        Remote IP addr                          : 0.0.0.0

FEC mode                                        : Soft-Decision 15

AINS Soak                                       : None
AINS Timer                                      : 0h, 0m
AINS remaining time                             : 0 seconds
```

# Regeneration Mode

In an optical transmission system, 3R regeneration helps extend the reach of the optical communication links by reamplifying, reshaping, and retiming the data pulses. Regeneration helps to correct any distortion of optical signals by converting it to an electrical signal, processing that electrical signal, and then retransmitting it again as an optical signal.

In Regeneration (Regen) mode, the OTN signal is received on a trunk port and the regenerated OTN signal is sent on the other trunk port of the line card and the other way round. In this mode, only the trunk optics controller and coherentDSP controllers are created.

## Configuring the Card in Regen Mode

The supported trunk rates for 1.2T card is100G to 600G in multiples of 100G.

To configure regen mode on 1.2T, use the following commands:

**configure**

**hw-module location** *location*

**regen**

**trunk-rate** *trunk-rate*

**commit**

**exit**

### Example

The following is a sample to configure the regen mode on 1.2T line card with the trunk-rate 300.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/0/NXR0
RP/0/RP0/CPU0:ios(config-hwmod)#regen
RP/0/RP0/CPU0:ios(config-regen)#trunk-rate 300
RP/0/RP0/CPU0:ios(config-regen)#commit
RP/0/RP0/CPU0:ios(config-regen)#exit
```

## Verifying the Regen Mode

The following is a sample to verify the regen mode.

**show hw-module location** *location* **regen**

```
RP/0/RP0/CPU0:ios#show hw-module location 0/0 regen
Mon Mar 25 09:50:42.936 UTC

Location:             0/0/NXR0
Trunk  Bitrate:       400G
Status:               Provisioned
East Port              West Port
CoherentDSP0/0/0/0      CoherentDSP0/0/0/1
```

The terms, East Port and West Port are used to represent OTN signal regeneration at the same layer.

# Configuring the BPS

You can configure the Bits per Symbol (BPS) to 3.4375 to support 300G trunk configurations on 75 GHz networks using the following commands:

**configure**

**controller optics** *R/S/I/P* **bits-per-symbol value**

**commit**

The following is a sample in which the BPS is configured to 3.4375.

```
RP/0/RP0/CPU0:ios#configure
Wed Mar 27 14:12:49.932 UTC
RP/0/RP0/CPU0:ios(config)#controller optics 0/3/0/0 bits-per-symbol 3.4375
RP/0/RP0/CPU0:ios(config)#commit
```

### Supported Baud Rates

*Table 1: Supported Baud Rates*

| Traffic Rate | Minimum Baud Rate | Maximum Baud Rate |
|---|---|---|
| 400 | 43.34518 | 130.4647 |
| 600 | 59.53435 | 148.0555 |
| 800 | 79.37913 | 148.0555 |
| 1000 | 99.22392 | 148.0555 |

### Viewing BPS and Baud Rate Ranges

To view the the BPS for a specific range use the following command:

**show controller optics** *R/S/I/P* **bps-range** *bps-range* **| include** *data-rate* **| include** *fec-type*

```
RP/0/RP0/CPU0:ios#show controllers optics 0/3/0/0 bps-range 3 3.05 | include 300G | include
 SD27
Thu Mar 28 03:01:39.751 UTC
300G            SD27            3.0000000       69.4350994
300G            SD27            3.0078125       69.2547485
300G            SD27            3.0156250       69.0753320
300G            SD27            3.0234375       68.8968428
300G            SD27            3.0312500       68.7192736
300G            SD27            3.0390625       68.5426174
300G            SD27            3.0468750       68.3668671
```

To view the baud for a specific range use the following command:

**show controller optics** *R/S/I/P* **baud-rate-range** *baud-range* **| include** *data-rate* **| include** *fec-type*

```
RP/0/RP0/CPU0:ios#show controllers optics 0/3/0/0 baud-rate-range 43 43.4 | include 300G |
 include SD27
Thu Mar 28 03:12:36.521 UTC
300G            SD27            4.8046875       43.3545986
300G            SD27            4.8125000       43.2842178
300G            SD27            4.8203125       43.2140651
300G            SD27            4.8281250       43.1441394
```

```
300G            SD27            4.8359375       43.0744397
300G            SD27            4.8437500       43.0049648
```

# Configuring the Trunk Rate for BPSK

You can configure trunk rates of 50G, 100G, and 150G to support Binary Phase-Shift Keying (BPSK) modulation. The BPSK modulation enables information to be carried over radio signals more efficiently.

You can configure trunk rates for BPSK using CLI, NetConf YANG, and OC models.

The following table list the 50G, 100G, and 150G trunk rates with the supported BPSK modulation:

| Trunk Rate | BPSK Modulation |
|---|---|
| 50G | 1 to 1.4453125 |
| 100G | 1 to 2.890625 |
| 150G | 1.453125 to 4.3359375 |

To configure the trunk rate for BPSK modulation, enter the following commands:

**configure**

**hw-module location** *location* **mxponder**

**trunk-rate {50G | 100G | 150G}**

**commit**

The following example shows how to configure trunk rate to 50G:

```
RP/0/RP0/CPU0:(config)#hw-module location 0/0/NXR0 mxponder
RP/0/RP0/CPU0:(config-hwmod-mxp)#trunk-rate 50G
RP/0/RP0/CPU0:(config-hwmod-mxp)#commit
```

## Viewing the BPSK Trunk Rate Ranges

To view the trunk rate configured for the BPSK modulation, use the following **show** commands:

```
RP/0/RP0/CPU0:ios(hwmod-mxp)#show hw-module location 0/0/NXR0 mxponder

Tue Feb 25 11:13:41.934 UTC


Location:             0/0/NXR0
Client Bitrate:       100GE
Trunk  Bitrate:       50G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/0/0/0
CoherentDSP0/0/0/1
                                Traffic Split Percentage


HundredGigECtrlr0/0/0/2         ODU40/0/0/0                            50
      50
```

```
RP/0/RP0/CPU0:ios#show controllers optics 0/0/0/0
Thu Mar  5 07:12:55.681 UTC

Controller State: Up

Transport Admin State: In Service

Laser State: On

LED State: Green

 Optics Status

         Optics Type:  DWDM optics
         DWDM carrier Info: C BAND, MSA ITU Channel=61, Frequency=193.10THz,
         Wavelength=1552.524nm

         Alarm Status:
         -------------
         Detected Alarms: None


         LOS/LOL/Fault Status:

         Alarm Statistics:

         -------------
         HIGH-RX-PWR = 0            LOW-RX-PWR = 2
         HIGH-TX-PWR = 0            LOW-TX-PWR = 0
         HIGH-LBC = 0               HIGH-DGD = 0
         OOR-CD = 0                 OSNR = 0
         WVL-OOL = 0                MEA  = 0
         IMPROPER-REM = 0
         TX-POWER-PROV-MISMATCH = 0
         Laser Bias Current = 0.0 %
         Actual TX Power = 1.97 dBm
         RX Power = 1.58 dBm
         RX Signal Power = 0.60 dBm
         Frequency Offset = 386 MHz

         Performance Monitoring: Enable

         THRESHOLD VALUES
         ----------------

         Parameter                 High Alarm  Low Alarm  High Warning  Low Warning
         ------------------------  ----------  ---------  ------------  -----------
         Rx Power Threshold(dBm)          4.9      -12.0           0.0          0.0
         Tx Power Threshold(dBm)          3.5      -10.1           0.0          0.0
         LBC Threshold(mA)                N/A        N/A          0.00         0.00

         Configured Tx Power = 2.00 dBm
         Configured CD High Threshold = 180000 ps/nm
         Configured CD lower Threshold = -180000 ps/nm
         Configured OSNR lower Threshold = 0.00 dB
         Configured DGD Higher Threshold = 180.00 ps
         Baud Rate =  34.7175521851 GBd
         Bits per Symbol = 1.0000000000  bits/symbol
         Modulation Type: BPSK
         Chromatic Dispersion -9 ps/nm
         Configured CD-MIN -180000 ps/nm  CD-MAX 180000 ps/nm
         Polarization Mode Dispersion = 0.0 ps
         Second Order Polarization Mode Dispersion = 125.00 ps^2
         Optical Signal to Noise Ratio = 34.60 dB
         SNR = 20.30 dB
         Polarization Dependent Loss = 0.20 dB
         Polarization Change Rate = 0.00 rad/s
         Differential Group Delay = 2.00 ps
         Filter Roll Off Factor : 0.100
         Rx VOA Fixed Ratio : 15.00 dB
         Enhanced Colorless Mode : 0
         Enhanced SOP Tolerance Mode : 0
         NLEQ Compensation Mode : 0
         Cross Polarization Gain Mode : 0
         Cross Polarization Weight Mode : 0
         Carrier Phase Recovery Window : 0
         Carrier Phase Recovery Extended Window : 0

AINS Soak                : None
AINS Timer               : 0h, 0m
AINS remaining time      : 0 seconds
```

# 2.4T Line Card

The following section describes the supported configurations and procedures to configure the card modes on the 2.4T line card.

## Card Modes

The line card supports the muxponder slice configurations.

The line cards have two trunk ports (0 and 7) and 6 client ports (from 1 to 6) each. You can configure the line card in the following mode:

- Muxponder slice—In this mode, each trunk port is configured independent of the other with different trunk rates. The client-to-trunk mapping is fixed. For Trunk 0, the client ports are 1 to 3. For Trunk 7, the client ports are 4 to 6.

## Supported Data Rates

The following data rates are supported on the 2.4T line card.

- 400G
- 600G
- 800G
- 1000G

### Muxponder Slice Data Rates

The following table displays the client and trunk ports that are enabled for the muxponder slice 0 (Trunk 0) configuration.

*Table 2: Slice 0 Port Configurations*

| Trunk Data Rate | Supported Cards | Client Data Rate | Client Pluggable | Trunk Ports | Client Ports |
|---|---|---|---|---|---|
| Configure the 400G Trunk in the Muxponder Slice Mode | 2.4T | 400GE | QDD-400G[1] | 0 | 1 |
| | | 4x100GE | QDD-4x100G[2] | | |
| 5x100GE | 2xQDD-4x100G[2] | | | | |
| Configure the 600G Trunk in the Muxponder Slice Mode | 2.4T | 400GE+2x100GE | QDD-400G[1]+QDD-4x100G[2] | 0 | 1, 2 |
| | | 6x100GE | 2xQDD-4x100G[2] | | |
| Configure the 800G Trunk in the Muxponder Slice Mode | 2.4T | 2x400GE | 2xQDD-400G[1] | 0 | 1, 2 |
| | | 400GE + 4x100GE | QDD-400G[1]+QDD-4x100G[2] | | |
| | | 8x100GE | 2xQDD-4x100G[2] | | |
| Configure the 1000G Trunk in the Muxponder Slice Mode | 2.4T | 2x400GE + 2x100GE | 2xQDD-400G[1]+2xQDD-4x100G[2] | 0 | 1, 2, 3 |
| | | 10x100GE | 3xQDD-4x100G[2] | | |
| 2x400GE + 4x100GE | 2xQDD-400G[1]+QDD-4x100G[2] | | | | |
| 400GE + 8x100GE | QDD-400G[1]+2xQDD-4x100G[2] | | | | |
| 12x100GE | 3xQDD-4x100G[2] | | | | |

[1] QDD-400G refers to QDD-400G-FR4-S, QDD-400G-AOCxM, and QDD-400G-DR4-S pluggable modules.

[2] QDD-4x100G refers to QDD-4X100G-LR-S and QDD-400G-DR4-S pluggable modules.

The following table displays the client and trunk ports that are enabled for the muxponder slice 1 (Trunk 7) configuration.

*Table 3: Slice 1 Port Configurations*

| Trunk Data Rate | Card Support | Client Data Rate | Client Pluggable | Trunk Ports | Client Ports |
|---|---|---|---|---|---|
| Configure the 400G Trunk in the Muxponder Slice Mode | 2.4T | 400GE | QDD-400G[3] | 7 | 4 |
| | | 4x100GE | QDD-4x100G[4] | | |
| 5x100GE | 2xQDD-4x100G[4] | | | | |
| Configure the 600G Trunk in the Muxponder Slice Mode | 2.4T | 400GE + 2x100GE | QDD-400G[3]+QDD-4x100G[4] | 7 | 4, 5 |
| | | 6x100GE | 2xQDD-4x100G[4] | | |
| Configure the 800G Trunk in the Muxponder Slice Mode | 2.4T | 2x400GE | 2xQDD-400G[3] | 7 | 4, 5 |
| | | 400GE + 4x100GE | QDD-400G[3]+QDD-4x100G[4] | | |
| | | 8x100GE | 2xQDD-4x100G[4] | | |
| Configure the 1000G Trunk in the Muxponder Slice Mode | 2.4T | 2x400GE + 2x100GE | 2xQDD-400G[3]+QDD-4x100G[4] | 7 | 4, 5, 6 |
| | | 10x100GE | 3xQDD-4x100G[4] | | |
| 2x400GE + 4x100GE | 2xQDD-400G[3]+QDD-4x100G[4] | | | | |
| 400GE + 8x100GE | QDD-400G[3]+2x QDD-4x100G[4] | | | | |
| 12x100GE | 3xQDD-4x100G[4] | | | | |

[3] QDD-400G refers to QDD-400G-FR4-S, QDD-400G-AOCxM, and QDD-400G-DR4-S pluggable modules.

[4] QDD-4x100G refers to QDD-4X100G-LR-S and QDD-400G-DR4-S pluggable modules.

All configurations can be accomplished by using appropriate values for client bitrate and trunk bitrate parameters of the **hw-module** command.

The following table displays the baud rate ranges for each trunk rate in the 2.4T card.

*Table 4: 2.4T Card Baud Rate Ranges*

| Trunk Payload | Minimum Baud Rate (GBd) | Maximum Baud Rate (GBd) |
|---|---|---|
| 400G | 43.34518 | 130.4647 |
| 600G | 59.53435 | 148.0555 |

| Trunk Payload | Minimum Baud Rate (GBd) | Maximum Baud Rate (GBd) |
|---|---|---|
| 800G | 79.37913 | 148.0555 |
| 1000G | 99.22392 | 148.0555 |

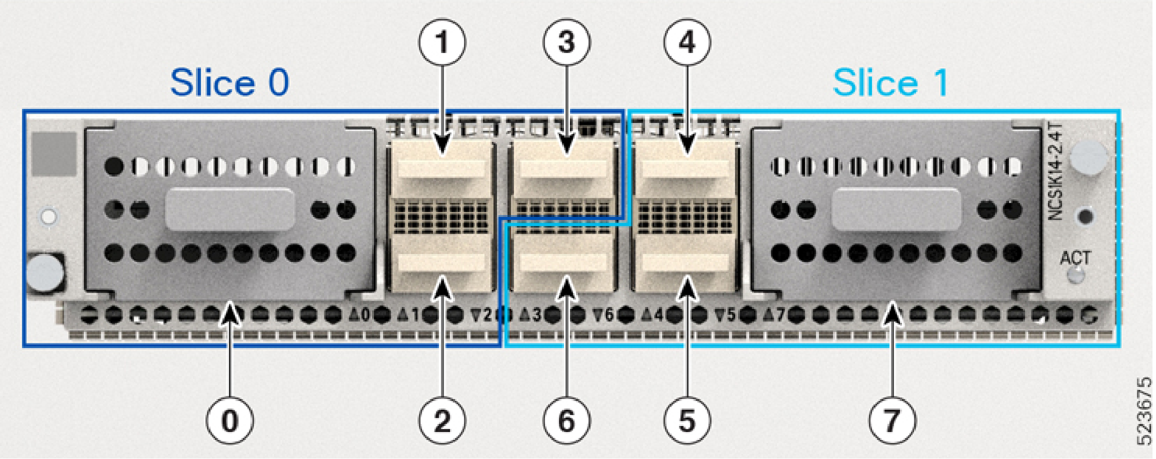
# Muxponder Slice Configuration on the 2.4T DWDM Card

You can configure the 2.4T DWDM Card in the slice configuration (muxponder slice) mode.

In the muxponder slice mode, the line card is divided into two slices, namely, Slice 0 and Slice 1. Each slice contains a trunk port and three client ports. In this mode, the trunk ports operate independently, carrying different data rates. The slices enable the card to function as two different modules. For example, if you set the trunk as 400G for Slice 0 and 600G for Slice 1, then Trunk 0 delivers 400G and Trunk 7 delivers 600G.

The following figure shows the line card slices and the respective trunk-to-client port mapping.

***Figure 1: 2.4T Line Card Slices and Ports***



| Slice 0 | | Slice 1 | |
|---|---|---|---|
| Trunk Port | Client Ports | Trunk Port | Client Ports |
| 0 | 1, 2, 3 | 7 | 4, 5, 6 |

To configure the card in the muxponder slice mode, use the following commands.

**configuration**

**hw-module location** *location* **mxponder-slice** *mxponder-slice-number*

**trunk-rate { 400G | 600G | 800G | 1000G }**

**client-port-rate** *client-port-number* **lane** *lane-number* **client-type 100GE**

**Note** QDD-4x100G-LR-S and QDD-400G-DR4-S client pluggable modules support both breakout and nonbreakout channels. For the 100GE breakout configuration, you must use the **lane** keyword. For the 400GE nonbreakout configuration, the **lane** keyword is not required.

**hw-module location** *location* **mxponder-slice** *mxponder-slice-number*

**trunk-rate { 400G | 600G | 800G | 1000G }**

**client-port-rate** *client-port-number* **client-type 400GE**

**commit**

**Note** Per trunk port, maximum time that it takes for the laser to be on after completion of datapath configuration is four minutes.

### Configure Muxponde Slice Mode on the 2.4T DWDM Card

The following sections describe the configurations for each trunk rate in the muxponde slice mode on the on the 2.4T DWDM card.

## Configure the 400G Trunk in the Muxponder Slice Mode

To configure the 2.4T card with 400G trunk rate in the muxponder slice mode, use the following commands:

**configuration**

**hw-module location** *location* **mxponder-slice** *mxponder-slice-number*

**trunk-rate 400G**

**client-port-rate** *client-port-number* **lane** *lane-number* **client-type 100GE** OR

**client-port-rate** *client-port-number* **client-type 400GE**

**commit**

**Examples**:

### Configuration 1: Using QDD-400G Pluggable

The following sample configuration provisions the 400G trunk rate on the *mxponder-slice 0* mode. This configuration has 400GE client on port 1.

**Example Commands**

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 0
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 400G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 1 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

The following sample configuration provisions the 400G trunk rate in the *mxponder-slice 1* mode. This configuration has 400GE client on port 4.

**Example Commands**

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 1
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 400G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

### Verify the 400G Slice Configuration

The following is a sample to verify the 400G trunk rate configured with 400GE client rate in the *mxponder-slice 0* mode.

#### Output Example

```
RP/0/RP0/CPU0:ios#show  hw-module location 0/1/NXR0 mxponder-slice 0
Thu Nov 16 15:41:25.720 UTC
Location:             0/1/NXR0
Slice ID:             0
Client Bitrate:       400GE
Trunk  Bitrate:       400G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/0
                                Traffic Split Percentage

FourHundredGigECtrlr0/1/0/1     ODU-FLEX0/1/0/0/1                             100
```

The following is a sample to verify the 400G trunk rate configured with 400GE client rate in the *mxponder-slice 1* mode.

#### Output Example

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 1
Thu Nov 16 16:00:52.022 UTC

Location:             0/1/NXR0
Slice ID:             1
Client Bitrate:       400GE
Trunk  Bitrate:       400G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/7
                                Traffic Split Percentage

FourHundredGigECtrlr0/1/0/4     ODU-FLEX0/1/0/7/4                             100
```

### Configuration 2: Using QDD-4x100G Pluggable Module

The following sample configuration provisions the 400G trunk rate in the *mxponder-slice 0* mode. This configuration has 4x100GE client on port 1.

#### Example Commands

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 0
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 400G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 1 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 1 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 1 lane 3 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 1 lane 4 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

The following sample configuration provisions the 400G trunk rate in the *mxponder-slice 1* mode. This configuration has 4x100GE client on port 4.

### Example Commands

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 1
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 400G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 4 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

## Verify the 400G Slice Configuration

The following is a sample to verify the 400G trunk rate configured with 100GE client rate in the *mxponder-slice 0* mode.

### Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 0
Thu Nov 16 16:06:57.575 UTC

Location:             0/1/NXR0
Slice ID:             0
Client Bitrate:       100GE
Trunk  Bitrate:       400G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/0
                                Traffic Split Percentage

HundredGigECtrlr0/1/0/1/1       ODU-FLEX0/1/0/0/1/1                          100
HundredGigECtrlr0/1/0/1/2       ODU-FLEX0/1/0/0/1/2                          100
HundredGigECtrlr0/1/0/1/3       ODU-FLEX0/1/0/0/1/3                          100
HundredGigECtrlr0/1/0/1/4       ODU-FLEX0/1/0/0/1/4                          100
```

The following is a sample to verify the 400G trunk rate configured with 100GE client rate in the *mxponder-slice 1* mode.

### Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 1
Thu Nov 16 16:06:57.575 UTC

Location:             0/1/NXR0
Slice ID:             1
Client Bitrate:       100GE
Trunk  Bitrate:       400G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/7
                                Traffic Split Percentage

HundredGigECtrlr0/1/7/4/1       ODU-FLEX0/1/0/7/4/1                          100
HundredGigECtrlr0/1/7/4/2       ODU-FLEX0/1/0/7/4/2                          100
HundredGigECtrlr0/1/7/4/3       ODU-FLEX0/1/0/7/4/3                          100
HundredGigECtrlr0/1/7/4/4       ODU-FLEX0/1/0/7/4/4                          100
```

# Configure the 600G Trunk in the Muxponder Slice Mode

To configure the 2.4T card with 600G trunk rate in the muxponder slice mode, use the following commands:

**configuration**

**hw-module location** *location* **mxponder-slice** *mxponder-slice-number*

**trunk-rate 600G**

**client-port-rate** *client-port-number* **lane** *lane-number* **client-type 100GE**

**client-port-rate** *client-port-number* **client-type 400GE**

**commit**

### Configuration 1: Using QDD-400G and QDD-4x100G Pluggable Modules

The following sample configuration provisions the 2.4T card with the 600G trunk rate in the *mxponder-slice 0* mode. This configuration has 400GE client on port 1 and 2x100GE client on port 2.

**Example Commands**

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 0
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 600G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 1 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 2 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 2 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

The following sample configuration provisions the 2.4T card with the 600G trunk rate in the *mxponder-slice 1* mode. This configuration has 400GE client on port 4 and 2x100GE client on port 5.

**Example Commands**

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 1
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 600G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

### Verify the 600G Slice Configuration

The following is a sample to verify the 600G trunk rate configured with mixed client rate in the *mxponder-slice 0* mode.

**Example Output**

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 0
Thu Nov 16 16:20:04.745 UTC

Location:             0/1/NXR0
Slice ID:             0
Client Bitrate:       MIXED
Trunk  Bitrate:       600G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/0
                                Traffic Split Percentage
```

```
FourHundredGigECtrlr0/1/0/1     ODU-FLEX0/1/0/0/1                             100
HundredGigECtrlr0/1/0/2/1       ODU-FLEX0/1/0/0/2/1                           100
HundredGigECtrlr0/1/0/2/2       ODU-FLEX0/1/0/0/2/2                           100
```

The following is a sample to verify the 600G trunk rate configured with mixed client rate in the *mxponder-slice 1* mode.

### Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 1
Thu Nov 16 16:20:04.745 UTC

Location:             0/1/NXR0
Slice ID:             1
Client Bitrate:        MIXED
Trunk  Bitrate:       600G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/7
                                Traffic Split Percentage

FourHundredGigECtrlr0/1/0/4     ODU-FLEX0/1/0/7/4                             100
HundredGigECtrlr0/1/0/5/1       ODU-FLEX0/1/0/7/5/1                           100
HundredGigECtrlr0/1/0/5/2       ODU-FLEX0/1/0/7/5/2                           100
```

## Configuration 2: Using QDD-4x100G Pluggable Module

The following sample configuration provisions the 2.4T card with the 600G trunk rate in the *mxponder-slice 0* mode. This configuration has 4x100GE client on ports 1 and 2.

### Example Commands

```
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0
RP/0/RP0/CPU0:ios(config-hwmod)#mxponder-slice 0
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 600G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 1 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 1 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 1 lane 3 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 1 lane 4 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 2 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 2 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

The following sample configuration provisions the 2.4T card with the 600G trunk rate in the *mxponder-slice 0* mode. This configuration has 4x100GE client on ports 4 and 5.

### Example Commands

```
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0
RP/0/RP0/CPU0:ios(config-hwmod)#mxponder-slice 1
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 600G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 3 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 4 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

### Verify the 600G Slice Configuration

The following is a sample to verify the 600G trunk rate configured with 100GE client in the *mxponder-slice 0* mode.

#### Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 0
Thu Nov 16 16:35:23.121 UTC

Location:             0/1/NXR0
Slice ID:             0
Client Bitrate:       100GE
Trunk  Bitrate:       600G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/0
                                Traffic Split Percentage

HundredGigECtrlr0/1/0/1/1       ODU-FLEX0/1/0/0/1/1                           100
HundredGigECtrlr0/1/0/1/2       ODU-FLEX0/1/0/0/1/2                           100
HundredGigECtrlr0/1/0/1/3       ODU-FLEX0/1/0/0/1/3                           100
HundredGigECtrlr0/1/0/1/4       ODU-FLEX0/1/0/0/1/4                           100
HundredGigECtrlr0/1/0/2/1       ODU-FLEX0/1/0/0/2/1                           100
HundredGigECtrlr0/1/0/2/2       ODU-FLEX0/1/0/0/2/2                           100
```

The following is a sample to verify the 600G trunk rate configured with 100GE client rate in the *mxponder-slice 1* mode.

#### Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 1
Thu Nov 16 16:35:23.121 UTC

Location:             0/1/NXR0
Slice ID:             1
Client Bitrate:       100GE
Trunk  Bitrate:       600G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/7
                                Traffic Split Percentage

HundredGigECtrlr0/1/0/4/1       ODU-FLEX0/1/0/7/4/1                           100
HundredGigECtrlr0/1/0/4/2       ODU-FLEX0/1/0/7/4/2                           100
HundredGigECtrlr0/1/0/4/3       ODU-FLEX0/1/0/7/4/3                           100
HundredGigECtrlr0/1/0/4/4       ODU-FLEX0/1/0/7/4/4                           100
HundredGigECtrlr0/1/0/5/1       ODU-FLEX0/1/0/7/5/1                           100
HundredGigECtrlr0/1/0/5/2       ODU-FLEX0/1/0/7/5/2                           100
```

## Configure the 800G Trunk in the Muxponder Slice Mode

To configure the 2.4T card with 800G trunk rate in the muxponder slice mode, use the following commands:

**configuration**

**hw-module location** *location* **mxponder-slice** *mxponder-slice-number*

**trunk-rate 800G**

**client-port-rate** *client-port-number* **lane** *lane-number* **client-type 100GE**

**client-port-rate** *client-port-number* **client-type 400GE**

**commit**

### Configuration 1: Using QDD-400G and QDD-4x100G Pluggable Modules

The following sample configuration provisions the 2.4T card with the 800G trunk rate in the *mxponder-slice 0* mode. This configuration has 400GE client on port 1 and 4x100GE client on port 2.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 0
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 800G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 1 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 2 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 2 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 2 lane 3 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 2 lane 4 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

The following sample configuration provisions the 2.4T card with the 800G trunk rate in the *mxponder-slice 1* mode. This configuration has 400GE client on port 4 and 4x100GE client on port 5.

#### Example Commands

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 1
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 800G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 3 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 4 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

### Verify the 800G Slice Configuration

The following is a sample to verify the 800G trunk rate configured with mixed client rates in the *mxponder-slice 0* mode.

#### Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 0
Thu Nov 16 15:17:14.082 UTC

Location:             0/1/NXR0
Slice ID:             0
Client Bitrate:       MIXED
Trunk  Bitrate:       800G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/0
                                Traffic Split Percentage

HundredGigECtrlr0/1/0/2/1       ODU-FLEX0/1/0/0/2/1                           100
HundredGigECtrlr0/1/0/2/2       ODU-FLEX0/1/0/0/2/2                           100
HundredGigECtrlr0/1/0/2/3       ODU-FLEX0/1/0/0/2/3                           100
HundredGigECtrlr0/1/0/2/4       ODU-FLEX0/1/0/0/2/4                           100
FourHundredGigECtrlr0/1/0/1     ODU-FLEX0/1/0/0/1                             100
```

The following is a sample to verify the 800G trunk rate configured with mixed client rates in the *mxponder-slice 1* mode.

### Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 1
Thu Nov 16 15:20:51.482 UTC

Location:             0/1/NXR0
Slice ID:             1
Client Bitrate:       MIXED
Trunk  Bitrate:       800G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/7
                                Traffic Split Percentage

HundredGigECtrlr0/1/0/5/1       ODU-FLEX0/1/0/7/5/1                           100
HundredGigECtrlr0/1/0/5/2       ODU-FLEX0/1/0/7/5/2                           100
HundredGigECtrlr0/1/0/5/3       ODU-FLEX0/1/0/7/5/3                           100
HundredGigECtrlr0/1/0/5/4       ODU-FLEX0/1/0/7/5/4                           100
FourHundredGigECtrlr0/1/0/4     ODU-FLEX0/1/0/7/4                             100
```

## Configuration 2: Using QDD-400G Pluggable Module

The following sample configuration provisions the 2.4T card with the 800G trunk rate on the *mxponder-slice 0* mode. This configuration has 400GE client on ports 1and 2.

### Example Commands

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 0
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 800G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 1 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 2 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

The following sample configuration provisions the 2.4T card with the 800G trunk rate in the *mxponder-slice 1* mode. This configuration has 400GE client on ports 4 and 5.

### Example Commands

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 1
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 800G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

## Verify the 800G Slice Configuration

The following is a sample to verify the 800G trunk rate configured with 400GE client rate in the *mxponder-slice 0* mode.

### Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 0
Thu Nov 16 13:51:31.599 UTC

Location:             0/1/NXR0
Slice ID:             0
Client Bitrate:       400GE
Trunk  Bitrate:       800G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
```

```
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/0
                                Traffic Split Percentage

FourHundredGigECtrlr0/1/0/1     ODU-FLEX0/1/0/0/1                            100
FourHundredGigECtrlr0/1/0/2     ODU-FLEX0/1/0/0/2                            100
```

The following is a sample to verify the 800G trunk rate configured with 400GE client rate in the *mxponder-slice 1* mode.

### Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 1
Thu Nov 16 14:50:32.407 UTC

Location:             0/1/NXR0
Slice ID:             1
Client Bitrate:       400GE
Trunk  Bitrate:       800G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/7
                                Traffic Split Percentage

FourHundredGigECtrlr0/1/0/4     ODU-FLEX0/1/0/7/4                            100
FourHundredGigECtrlr0/1/0/5     ODU-FLEX0/1/0/7/5                            100
```

## Configuration 3: Using 4x100GE Pluggable

The following sample configuration provisions the 2.4T card with the 800G trunk rate in the *mxponder-slice 0* mode. This configuration has 4x100GE client on ports 1 and 2.

### Example Commands

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0
RP/0/RP0/CPU0:ios(config-hwmod)#mxponder-slice 0
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 800G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 1 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 1 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 1 lane 3 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 1 lane 4 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 2 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 2 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 2 lane 3 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 2 lane 4 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

The following sample configuration provisions the 2.4T card with the 800G trunk rate in the *mxponder-slice 1* mode. This configuration has 4x100GE client on ports 4 and 5.

### Example Commands

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0
RP/0/RP0/CPU0:ios(config-hwmod)#mxponder-slice 1
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 800G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 3 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 4 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 1 client-type 100GE
```

```
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 3 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 4 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
Thu Nov 16 15:01:28.989 UTC
```

### Verify the 800G Slice Configuration

The following is a sample to verify the 800G trunk rate configured with 100GE client rate in the *mxponder-slice 0* mode.

#### Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 0
Thu Nov 16 14:59:39.538 UTC

Location:             0/1/NXR0
Slice ID:             0
Client Bitrate:       100GE
Trunk  Bitrate:       800G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/0
                                Traffic Split Percentage

HundredGigECtrlr0/1/0/1/1       ODU-FLEX0/1/0/0/1/1                           100
HundredGigECtrlr0/1/0/1/2       ODU-FLEX0/1/0/0/1/2                           100
HundredGigECtrlr0/1/0/1/3       ODU-FLEX0/1/0/0/1/3                           100
HundredGigECtrlr0/1/0/1/4       ODU-FLEX0/1/0/0/1/4                           100
HundredGigECtrlr0/1/0/2/1       ODU-FLEX0/1/0/0/2/1                           100
HundredGigECtrlr0/1/0/2/2       ODU-FLEX0/1/0/0/2/2                           100
HundredGigECtrlr0/1/0/2/3       ODU-FLEX0/1/0/0/2/3                           100
HundredGigECtrlr0/1/0/2/4       ODU-FLEX0/1/0/0/2/4                           100
```

The following is a sample to verify the 800G trunk rate configured with 100GE client rate in the *mxponder-slice 1* mode.

#### Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 1
Thu Nov 16 15:11:45.618 UTC

Location:             0/1/NXR0
Slice ID:             1
Client Bitrate:       100GE
Trunk  Bitrate:       800G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/7
                                Traffic Split Percentage

HundredGigECtrlr0/1/0/4/1       ODU-FLEX0/1/0/7/4/1                           100
HundredGigECtrlr0/1/0/4/2       ODU-FLEX0/1/0/7/4/2                           100
HundredGigECtrlr0/1/0/4/3       ODU-FLEX0/1/0/7/4/3                           100
HundredGigECtrlr0/1/0/4/4       ODU-FLEX0/1/0/7/4/4                           100
HundredGigECtrlr0/1/0/5/1       ODU-FLEX0/1/0/7/5/1                           100
HundredGigECtrlr0/1/0/5/2       ODU-FLEX0/1/0/7/5/2                           100
HundredGigECtrlr0/1/0/5/3       ODU-FLEX0/1/0/7/5/3                           100
HundredGigECtrlr0/1/0/5/4       ODU-FLEX0/1/0/7/5/4                           100
```

## Configure the 1000G Trunk in the Muxponder Slice Mode

To configure the 2.4T card with the 1000G trunk rate in the muxponder slice mode, use the following commands:

**configuration**

**hw-module location** *location* **mxponder-slice** *mxponder-slice-number*

**trunk-rate 1000G**

**client-port-rate** *client-port-number* **lane** *lane-number* **client-type 100GE**

**client-port-rate** *client-port-number* **client-type 400GE**

**commit**

### Configuration 1: Using QDD-400G and QDD-4x100GE Pluggable Modules

The following sample configuration provisions the 2.4T card with the 1000G trunk rate in the *mxponder-slice 0* mode. This configuration has 400GE client on ports 1, 2 and 2x100GE client on port 3.

**Example Commands**

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 0
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 1000G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 1 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 2 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 3 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 3 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

The following sample configuration provisions the 2.4T card with the 1000G trunk rate in the *mxponder-slice 1* mode. This configuration has 400GE client on ports 4, 5 and 2x100GE client on port 6.

**Example Commands**

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0 mxponder-slice 1
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 1000G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 6 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 6 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

### Verify the 1000G Slice Configuration

The following is a sample to verify the 1000G configured with mixed client rate in the *mxponder-slice 0* mode.

**Example Output**

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 0
Thu Nov 16 15:17:14.082 UTC

Location:             0/1/NXR0
Slice ID:             0
Client Bitrate:       MIXED
Trunk  Bitrate:       1000G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
```

```
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/0
                                Traffic Split Percentage

HundredGigECtrlr0/1/0/3/1       ODU-FLEX0/1/0/0/3/1                             100
HundredGigECtrlr0/1/0/3/2       ODU-FLEX0/1/0/0/3/2                             100
FourHundredGigECtrlr0/1/0/1     ODU-FLEX0/1/0/0/1                              100
FourHundredGigECtrlr0/1/0/2     ODU-FLEX0/1/0/0/2                              100
```

The following is a sample to verify the 1000G trunk rate configured with mixed client rate in the *mxponder-slice 1* mode.

### Example Output

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 1
Thu Nov 16 15:20:51.482 UTC

Location:             0/1/NXR0
Slice ID:             1
Client Bitrate:       MIXED
Trunk  Bitrate:       1000G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/7
                                Traffic Split Percentage

HundredGigECtrlr0/1/0/6/1       ODU-FLEX0/1/0/7/6/1                             100
HundredGigECtrlr0/1/0/6/2       ODU-FLEX0/1/0/7/6/2                             100
FourHundredGigECtrlr0/1/0/4     ODU-FLEX0/1/0/7/4                               100
FourHundredGigECtrlr0/1/0/5     ODU-FLEX0/1/0/7/5                               100
```

## Configuration 2: Using QDD-4x100G Pluggable Module

The following sample configuration provisions the 2.4T card with the 1000G trunk rate in the *mxponder-slice 0* mode. This configuration has 4x100GE client on ports 1, 2, and 3.

### Example Commands

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0
RP/0/RP0/CPU0:ios(config-hwmod)#mxponder-slice 0
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 1000G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 1 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 1 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 1 lane 3 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 1 lane 4 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 2 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 2 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 2 lane 3 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 2 lane 4 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 3 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 3 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
```

The following sample configuration provisions the 2.4T card with the 1000G trunk rate in the *mxponder-slice 1* mode. This configuration has 4x100GE client on ports 4, 5, and 6.

### Example Commands

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1/NXR0
RP/0/RP0/CPU0:ios(config-hwmod)#mxponder-slice 1
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 1000G
```

```
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 3 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 4 lane 4 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 3 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 5 lane 4 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 6 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#client-port-rate 6 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#commit
Thu Nov 16 15:01:28.989 UTC
```

### Verify the 1000G Slice Configuration

The following is a sample to verify the 1000G trunk rate configured with 100GE client rate in the *mxponder-slice 0* mode.

**Example Output**

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 0
Thu Nov 16 14:59:39.538 UTC

Location:             0/1/NXR0
Slice ID:             0
Client Bitrate:       100GE
Trunk  Bitrate:       1000G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/0
                                Traffic Split Percentage

HundredGigECtrlr0/1/0/1/1       ODU-FLEX0/1/0/0/1/1                           100
HundredGigECtrlr0/1/0/1/2       ODU-FLEX0/1/0/0/1/2                           100
HundredGigECtrlr0/1/0/1/3       ODU-FLEX0/1/0/0/1/3                           100
HundredGigECtrlr0/1/0/1/4       ODU-FLEX0/1/0/0/1/4                           100
HundredGigECtrlr0/1/0/2/1       ODU-FLEX0/1/0/0/2/1                           100
HundredGigECtrlr0/1/0/2/2       ODU-FLEX0/1/0/0/2/2                           100
HundredGigECtrlr0/1/0/2/3       ODU-FLEX0/1/0/0/2/3                           100
HundredGigECtrlr0/1/0/2/4       ODU-FLEX0/1/0/0/2/4                           100
HundredGigECtrlr0/1/0/3/1       ODU-FLEX0/1/0/0/3/1                           100
HundredGigECtrlr0/1/0/3/2       ODU-FLEX0/1/0/0/3/2                           100
```

The following is a sample to verify the 1000G trunk rate configure with 100GE client rate configured in the *mxponder-slice 1* mode.

**Example Output**

```
RP/0/RP0/CPU0:ios#show hw-module location 0/1/NXR0 mxponder-slice 1
Thu Nov 16 15:11:45.618 UTC

Location:             0/1/NXR0
Slice ID:             1
Client Bitrate:       100GE
Trunk  Bitrate:       1000G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port                     Mapper/Trunk Port          CoherentDSP0/1/0/7
                                Traffic Split Percentage

HundredGigECtrlr0/1/0/4/1       ODU-FLEX0/1/0/7/4/1                           100
```

```
HundredGigECtrlr0/1/0/4/2       ODU-FLEX0/1/0/7/4/2                          100
HundredGigECtrlr0/1/0/4/3       ODU-FLEX0/1/0/7/4/3                          100
HundredGigECtrlr0/1/0/4/4       ODU-FLEX0/1/0/7/4/4                          100
HundredGigECtrlr0/1/0/5/1       ODU-FLEX0/1/0/7/5/1                          100
HundredGigECtrlr0/1/0/5/2       ODU-FLEX0/1/0/7/5/2                          100
HundredGigECtrlr0/1/0/5/3       ODU-FLEX0/1/0/7/5/3                          100
HundredGigECtrlr0/1/0/5/4       ODU-FLEX0/1/0/7/5/4                          100
HundredGigECtrlr0/1/0/6/1       ODU-FLEX0/1/0/7/6/1                          100
HundredGigECtrlr0/1/0/6/2       ODU-FLEX0/1/0/7/6/2                          100
```