



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, on page 1](#)
- [2.4T and 2.4TX Line Cards, on page 16](#)
- [QXP Card, on page 43](#)

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 rates in 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 output 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 output 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 output 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

HundredGigEctrler0/1/0/2   ODU40/1/0/0/1           100           0
HundredGigEctrler0/1/0/3   ODU40/1/0/0/2           100           0
HundredGigEctrler0/1/0/4   ODU40/1/0/0/3           100           0
HundredGigEctrler0/1/0/5   ODU40/1/0/0/4           100           0
HundredGigEctrler0/1/0/6   ODU40/1/0/0/5           100           0
HundredGigEctrler0/1/0/7   ODU40/1/0/0/6           50            50
HundredGigEctrler0/1/0/8   ODU40/1/0/1/1           0            100
HundredGigEctrler0/1/0/9   ODU40/1/0/1/2           0            100
HundredGigEctrler0/1/0/10  ODU40/1/0/1/3           0            100
HundredGigEctrler0/1/0/11  ODU40/1/0/1/4           0            100
HundredGigEctrler0/1/0/12  ODU40/1/0/1/5           0            100
    
```

The following is a sample output 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

HundredGigEctrler0/1/0/2   ODU40/1/0/0/1           100
HundredGigEctrler0/1/0/3   ODU40/1/0/0/2           100
HundredGigEctrler0/1/0/4   ODU40/1/0/0/3           100
HundredGigEctrler0/1/0/5   ODU40/1/0/0/4           100
HundredGigEctrler0/1/0/6   ODU40/1/0/0/5           100
    
```

The following is a sample output 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

HundredGigEctrler0/1/0/8   ODU40/1/0/1/1           100
HundredGigEctrler0/1/0/9   ODU40/1/0/1/2           100
HundredGigEctrler0/1/0/10  ODU40/1/0/1/3           100
HundredGigEctrler0/1/0/11  ODU40/1/0/1/4           100
    
```

The following is a sample output 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 is 100G 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-mpx)#trunk-rate 50G
RP/0/RP0/CPU0:(config-hwmod-mpx)#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-mpx)#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
 WV-L-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 and 2.4TX Line Cards

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

Card Modes

The line cards support muxponder slice and muxponder mode 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 modes:

- 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.
- Muxponder—In this mode, the card allows you to configure both trunk ports with the same trunk rate. The client-to-trunk mapping is fixed.



Note The muxponder mode is supported on the 2.4TX card only.

Supported Data Rates

The following data rates are supported on the 2.4T and 2.4TX line cards.

- 400G
- 500G
- 600G
- 800G

- 1000G
- 1200G

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, 2.4TX	400GE	QDD-400G ¹	0	1
		4x100GE	QDD-4x100G ²		
Configure the 500G Trunk in the Muxponder Slice Mode	2.4T, 2.4TX	400GE+1x100GE	QDD-400G ¹ + QDD-4x100G ²	0	1, 2
		5x100GE	2xQDD-4x100G ²		
Configure the 600G Trunk in the Muxponder Slice Mode	2.4T, 2.4TX	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, 2.4TX	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, 2.4TX	2x400GE + 2x100GE	2xQDD-400G ¹ +2xQDD-4x100G ²	0	1, 2, 3
		10x100GE	3xQDD-4x100G ²		
Configure the 1200G Trunk in the Muxponder Slice Mode	2.4T, 2.4TX	3x400GE	3xQDD-400G ¹	0	1, 2, 3
		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, 2.4TX	400GE	QDD-400G ³	7	4
		4x100GE	QDD-4x100G ⁴		
Configure the 500G Trunk in the Muxponder Slice Mode	2.4T, 2.4TX	400GE + 1x100GE	QDD-400G ³ +QDD-4x100G ⁴	7	4, 5
		5x100GE	2xQDD-4x100G ⁴		
Configure the 600G Trunk in the Muxponder Slice Mode	2.4T, 2.4TX	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, 2.4TX	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, 2.4TX	2x400GE + 2x100GE	2xQDD-400G ³ +QDD-4x100G ⁴	7	4, 5, 6
		10x100GE	3xQDD-4x100G ⁴		
Configure the 1200G Trunk in the Muxponder Slice Mode	2.4T, 2.4TX	3x400GE	3xQDD-400G ³	7	4, 5, 6
		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
500G	49.61196	147.7235
600G	59.53435	148.0555
800G	79.37913	148.0555
1000G	99.22392	148.0555
1200G	119.0687	148.0555

The following table displays the trunk parameter ranges for the 2.4TX card.

Table 5: 2.4TX Card Trunk Parameter Ranges

Trunk Payload	Minimum Baud Rate (GBd)	Maximum Baud Rate (GBd)	Default Baud Rate (GBd)	Minimum BPS	Maximum BPS
400G	43.34518	130.4647	127.931418	2.1	4.1
500G	49.61196	147.7235	137.8340588	2.5	5
600G	59.53435	148.0555	137.738007	2.8	5.1
800G	79.37913	148.0555	137.978388	3.5	5.1
1000G	99.22392	148.0555	137.834059	4.3	5.3
1200G	119.0687	148.0555	137.738007	5.3	5.7

Muxponder Mode Data Rates

The following table displays the data rates that are supported in the muxponder mode.

Table 6: Muxponder Mode Port Configurations

Trunk Rate	Total Configured Rate	Supported Cards	Trunk 0 Client Ports	Shared Client Port	Trunk 7 Client Ports
600G	1200G	2.4TX	1	2	4
1000G	2000G	2.4TX	1, 2	3	4, 5

Muxponder Slice Configuration on the 2.4T and 2.4TX Cards

You can configure the 2.4T and 2.4TX Cards 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

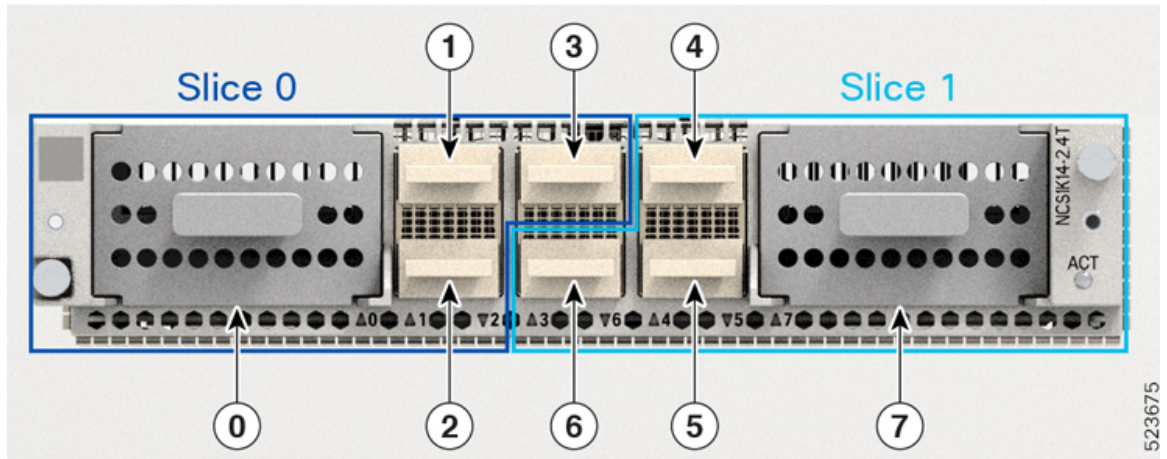
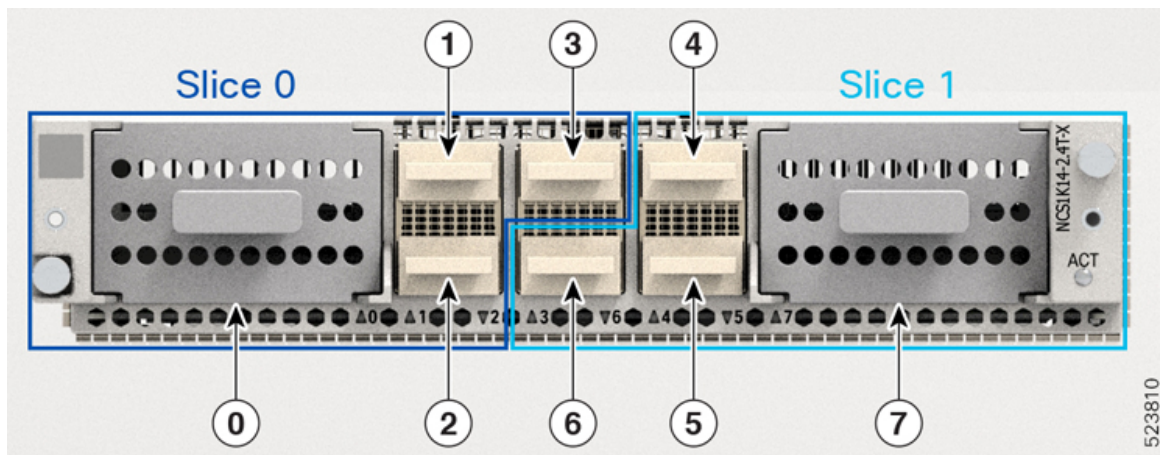


Figure 2: 2.4TX Line Card Slices and Ports



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 | 500G | 600G | 800G | 1000G | 1200G }

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 | 500G | 600G | 800G | 1000G | 1200G }
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 and 2.4TX Cards

The following sections describe the configurations for each trunk rate in the muxponde slice mode on the the 2.4T and 2.4TX cards.

Configure the 400G Trunk in the Muxponder Slice Mode

To configure the 2.4T and 2.4TX cards 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 muxponder-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 muxponder-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 muxponder-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 muxponder-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

HundredGigEctr1r0/1/0/1/1  ODU-FLEX0/1/0/0/1/1      100
HundredGigEctr1r0/1/0/1/2  ODU-FLEX0/1/0/0/1/2      100
HundredGigEctr1r0/1/0/1/3  ODU-FLEX0/1/0/0/1/3      100
HundredGigEctr1r0/1/0/1/4  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

HundredGigEctr1r0/1/7/4/1  ODU-FLEX0/1/0/7/4/1      100
HundredGigEctr1r0/1/7/4/2  ODU-FLEX0/1/0/7/4/2      100
HundredGigEctr1r0/1/7/4/3  ODU-FLEX0/1/0/7/4/3      100
HundredGigEctr1r0/1/7/4/4  ODU-FLEX0/1/0/7/4/4      100
```

Configure the 500G Trunk in the Muxponder Slice Mode

To configure the 2.4TX card with 500G trunk rate in the muxponder slice mode, use the following commands:

configuration

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

trunk-rate 500G

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 500G trunk rate in the *mxponder-slice 0* mode. This configuration has 400GE client on port 1 and 1x100GE 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 500G
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
```

The following sample configuration provisions the 500G trunk rate in the *mxponder-slice 1* mode. This configuration has 400GE client on port 4 and 1x100GE client on port 5.

```
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 500G
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
```

Verify the 500G Slice Configuration

The following is a sample to verify the 500G 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 15:41:25.720 UTC
Location:                0/1/NXR0
Slice ID:                 0
Client Bitrate:          MIXED
Trunk Bitrate:           500G
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
```

The following is a sample to verify the 500G 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 muxponder-slice 1
Thu Nov 16 16:00:52.022 UTC

Location:                0/1/NXR0
Slice ID:                 1
Client Bitrate:          MIXED
Trunk Bitrate:           500G
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
```

Configuration 2: Using QDD-4x100G Pluggable Module

The following sample configuration provisions the 2.4T card 2.4T and 2.4TX cards with the 500G 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)#muxponder-slice 0
RP/0/RP0/CPU0:ios (config-hwmod-mxp)#trunk-rate 500G
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)#commit
```

The following sample configuration provisions the 2.4T and 2.4TX cards with the 500G 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)#muxponder-slice 1
RP/0/RP0/CPU0:ios (config-hwmod-mxp)#trunk-rate 500G
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)#commit
```

Verify the 500G Slice Configuration

The following is a sample to verify the 500G 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 muxponder-slice 0
Thu Nov 16 16:35:23.121 UTC

Location:                0/1/NXR0
Slice ID:                 0
Client Bitrate:          100GE
Trunk Bitrate:           500G
Status:                  Provisioned
```

Configure the 600G Trunk in the Muxponder Slice Mode

```

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

HundredGigEctr1r0/1/0/1/1    ODU-FLEX0/1/0/0/1/1           100
HundredGigEctr1r0/1/0/1/2    ODU-FLEX0/1/0/0/1/2           100
HundredGigEctr1r0/1/0/1/3    ODU-FLEX0/1/0/0/1/3           100
HundredGigEctr1r0/1/0/1/4    ODU-FLEX0/1/0/0/1/4           100
HundredGigEctr1r0/1/0/2/1    ODU-FLEX0/1/0/0/2/1           100

```

The following is a sample to verify the 600G trunk rate configured with 100GE client rate in the *mxponder-slice /* 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:        500G
Status:               Provisioned
LLDP Drop Enabled:    FALSE
ARP Snoop Enabled:    FALSE
Client Port           Mapper/Trunk Port           CoherentDSP0/1/0/7
                    Traffic Split Percentage

HundredGigEctr1r0/1/0/4/1    ODU-FLEX0/1/0/7/4/1           100
HundredGigEctr1r0/1/0/4/2    ODU-FLEX0/1/0/7/4/2           100
HundredGigEctr1r0/1/0/4/3    ODU-FLEX0/1/0/7/4/3           100
HundredGigEctr1r0/1/0/4/4    ODU-FLEX0/1/0/7/4/4           100
HundredGigEctr1r0/1/0/5/1    ODU-FLEX0/1/0/7/5/1           100

```

Configure the 600G Trunk in the Muxponder Slice Mode

To configure the 2.4T and 2.4TX cards 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 and 2.4TX cards 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-mpx)#client-port-rate 1 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mpx)#client-port-rate 2 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mpx)#client-port-rate 2 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mpx)#commit
```

The following sample configuration provisions the 2.4T and 2.4TX cards 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-mpx)#trunk-rate 600G
RP/0/RP0/CPU0:ios(config-hwmod-mpx)#client-port-rate 4 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mpx)#client-port-rate 5 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mpx)#client-port-rate 5 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mpx)#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 and 2.4TX cards 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 and 2.4TX cards 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 muxponder-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

HundredGigEctr1r0/1/0/4/1  ODU-FLEX0/1/0/7/4/1      100
HundredGigEctr1r0/1/0/4/2  ODU-FLEX0/1/0/7/4/2      100
HundredGigEctr1r0/1/0/4/3  ODU-FLEX0/1/0/7/4/3      100
HundredGigEctr1r0/1/0/4/4  ODU-FLEX0/1/0/7/4/4      100
HundredGigEctr1r0/1/0/5/1  ODU-FLEX0/1/0/7/5/1      100
HundredGigEctr1r0/1/0/5/2  ODU-FLEX0/1/0/7/5/2      100
```

Configure the 800G Trunk in the Muxponder Slice Mode

To configure the 2.4T and 2.4TX cards 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 and 2.4TX cards 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 muxponder-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 and 2.4TX cards 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 muxponder-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 and 2.4TX cards with the 800G trunk rate on the *mxponder-slice 0* mode. This configuration has 400GE 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 muxponder-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 and 2.4TX cards 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 muxponder-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 muxponder-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 muxponder-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 and 2.4TX cards 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 and 2.4TX cards 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
```


HundredGigEctr1r0/1/0/1/3	ODU-FLEX0/1/0/0/1/3	100
HundredGigEctr1r0/1/0/1/4	ODU-FLEX0/1/0/0/1/4	100
HundredGigEctr1r0/1/0/2/1	ODU-FLEX0/1/0/0/2/1	100
HundredGigEctr1r0/1/0/2/2	ODU-FLEX0/1/0/0/2/2	100
HundredGigEctr1r0/1/0/2/3	ODU-FLEX0/1/0/0/2/3	100
HundredGigEctr1r0/1/0/2/4	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

HundredGigEctr1r0/1/0/4/1  ODU-FLEX0/1/0/7/4/1      100
HundredGigEctr1r0/1/0/4/2  ODU-FLEX0/1/0/7/4/2      100
HundredGigEctr1r0/1/0/4/3  ODU-FLEX0/1/0/7/4/3      100
HundredGigEctr1r0/1/0/4/4  ODU-FLEX0/1/0/7/4/4      100
HundredGigEctr1r0/1/0/5/1  ODU-FLEX0/1/0/7/5/1      100
HundredGigEctr1r0/1/0/5/2  ODU-FLEX0/1/0/7/5/2      100
HundredGigEctr1r0/1/0/5/3  ODU-FLEX0/1/0/7/5/3      100
HundredGigEctr1r0/1/0/5/4  ODU-FLEX0/1/0/7/5/4      100
```

Configure the 1000G Trunk in the Muxponder Slice Mode

To configure the 2.4T and 2.4TX cards 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 and 2.4TX cards 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-mpx)#client-port-rate 3 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mpx)#client-port-rate 3 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mpx)#commit
```

The following sample configuration provisions the 2.4T and 2.4TX cards 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-mpx)#trunk-rate 1000G
RP/0/RP0/CPU0:ios(config-hwmod-mpx)#client-port-rate 4 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mpx)#client-port-rate 5 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mpx)#client-port-rate 6 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mpx)#client-port-rate 6 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mpx)#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
```

FourHundredGigEctr1r0/1/0/4	ODU-FLEX0/1/0/7/4	100
FourHundredGigEctr1r0/1/0/5	ODU-FLEX0/1/0/7/5	100

Configuration 2: Using QDD-4x100G Pluggable Module

The following sample configuration provisions the 2.4T and 2.4TX cards 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 and 2.4TX cards 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

HundredGigEctr0/1/0/1/1    ODU-FLEX0/1/0/0/1/1    100
HundredGigEctr0/1/0/1/2    ODU-FLEX0/1/0/0/1/2    100
HundredGigEctr0/1/0/1/3    ODU-FLEX0/1/0/0/1/3    100
HundredGigEctr0/1/0/1/4    ODU-FLEX0/1/0/0/1/4    100
HundredGigEctr0/1/0/2/1    ODU-FLEX0/1/0/0/2/1    100
HundredGigEctr0/1/0/2/2    ODU-FLEX0/1/0/0/2/2    100
HundredGigEctr0/1/0/2/3    ODU-FLEX0/1/0/0/2/3    100
HundredGigEctr0/1/0/2/4    ODU-FLEX0/1/0/0/2/4    100
HundredGigEctr0/1/0/3/1    ODU-FLEX0/1/0/0/3/1    100
HundredGigEctr0/1/0/3/2    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

HundredGigEctr0/1/0/4/1    ODU-FLEX0/1/0/7/4/1    100
HundredGigEctr0/1/0/4/2    ODU-FLEX0/1/0/7/4/2    100
HundredGigEctr0/1/0/4/3    ODU-FLEX0/1/0/7/4/3    100
HundredGigEctr0/1/0/4/4    ODU-FLEX0/1/0/7/4/4    100
HundredGigEctr0/1/0/5/1    ODU-FLEX0/1/0/7/5/1    100
HundredGigEctr0/1/0/5/2    ODU-FLEX0/1/0/7/5/2    100
HundredGigEctr0/1/0/5/3    ODU-FLEX0/1/0/7/5/3    100
HundredGigEctr0/1/0/5/4    ODU-FLEX0/1/0/7/5/4    100
HundredGigEctr0/1/0/6/1    ODU-FLEX0/1/0/7/6/1    100
HundredGigEctr0/1/0/6/2    ODU-FLEX0/1/0/7/6/2    100

```

Configure the 1200G Trunk in the Muxponder Slice Mode

To configure the 2.4TX card with 1200G trunk rate in the muxponder slice mode, use the following commands:

configuration

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

trunk-rate 1200G

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 2.4TX card with the 1200G trunk rate in the *mxponder-slice 0* mode. This configuration has 400GE client on ports 1, 2 and 4x100GE client on port 3.

Example Commands

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/2/NXR0 mxponder-slice 0
RP/0/RP0/CPU0:ios(config-hwmod-mpx)#trunk-rate 1200G
RP/0/RP0/CPU0:ios(config-hwmod-mpx)#client-port-rate 1 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mpx)#client-port-rate 2 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mpx)#client-port-rate 3 lane 1 client-type 1x100GE
RP/0/RP0/CPU0:ios(config-hwmod-mpx)#client-port-rate 3 lane 2 client-type 1x100GE
RP/0/RP0/CPU0:ios(config-hwmod-mpx)#client-port-rate 3 lane 3 client-type 1x100GE
RP/0/RP0/CPU0:ios(config-hwmod-mpx)#client-port-rate 3 lane 4 client-type 1x100GE
```

The following sample configuration provisions the 2.4TX card with the 1200G trunk rate in the *mxponder-slice 1* mode. This configuration has 400GE client on ports 4, 5, and 4x100GE client on port 6.

Example Commands

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/2/NXR0 mxponder-slice 1
RP/0/RP0/CPU0:ios(config-hwmod-mpx)#trunk-rate 1100G
RP/0/RP0/CPU0:ios(config-hwmod-mpx)#client-port-rate 4 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mpx)#client-port-rate 5 client-type 400GE
RP/0/RP0/CPU0:ios(config-hwmod-mpx)#client-port-rate 6 lane 1 client-type 1x100GE
RP/0/RP0/CPU0:ios(config-hwmod-mpx)#client-port-rate 6 lane 2 client-type 1x100GE
RP/0/RP0/CPU0:ios(config-hwmod-mpx)#client-port-rate 6 lane 3 client-type 1x100GE
RP/0/RP0/CPU0:ios(config-hwmod-mpx)#client-port-rate 6 lane 4 client-type 1x100GE
```

Verify the 1200G Slice Configuration

The following is a sample to verify the 1200G trunk rate with mixed client rate configured 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:           1200G
Status:                  Provisioned
LLDP Drop Enabled:      FALSE
ARP Snoop Enabled:      FALSE
Client Port              Mapper/Trunk Port          CoherentDSP0/1/0/0
                          Traffic Split Percentage

HundredGigEctr1r0/1/0/3/1  ODU-FLEX0/1/0/0/3/1          100
HundredGigEctr1r0/1/0/3/2  ODU-FLEX0/1/0/0/3/2          100
HundredGigEctr1r0/1/0/3/3  ODU-FLEX0/1/0/0/3/3          100
HundredGigEctr1r0/1/0/3/4  ODU-FLEX0/1/0/0/3/4          100
FourHundredGigEctr1r0/1/0/1 ODU-FLEX0/1/0/0/1            100
FourHundredGigEctr1r0/1/0/2 ODU-FLEX0/1/0/0/2            100
```

The following is a sample to verify the 1200G trunk rate configured with mixed client rate in *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:           1200G
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
HundredGigECtrlr0/1/0/6/3    ODU-FLEX0/1/0/7/6/3    100
HundredGigECtrlr0/1/0/6/4    ODU-FLEX0/1/0/7/6/4    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-400G Pluggable Module

The following sample configuration provisions the 2.4TX card with the 1200G trunk rate in the *mxponder-slice 0* mode. This configuration has 400GE client on ports 1, 2, and 3.

Example Commands

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/2/NXR0 mxponder-slice 0
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 1200G
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 client-type 400GE
```

The following sample configuration provisions the 2.4TX card with the 1200G trunk rate in the *mxponder-slice 1* mode. This configuration has 400GE client on ports 4, 5, and 6.

Example Commands

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/2/NXR0 mxponder-slice 1
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 1200G
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 client-type 400GE
```

Verify the 1200G Configuration

The following is a sample to verify the 1200G 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:           1200G
Status:                  Provisioned
LLDP Drop Enabled:      FALSE
ARP Snoop Enabled:      FALSE
```

Client Port	Mapper/Trunk Port Traffic Split Percentage	CoherentDSP0/1/0/0
HundredGigEctr0/1/0/1/1	ODU-FLEX0/1/0/0/1/1	100
HundredGigEctr0/1/0/1/2	ODU-FLEX0/1/0/0/1/2	100
HundredGigEctr0/1/0/1/3	ODU-FLEX0/1/0/0/1/3	100
HundredGigEctr0/1/0/1/4	ODU-FLEX0/1/0/0/1/4	100
HundredGigEctr0/1/0/2/1	ODU-FLEX0/1/0/0/2/1	100
HundredGigEctr0/1/0/2/2	ODU-FLEX0/1/0/0/2/2	100
HundredGigEctr0/1/0/2/3	ODU-FLEX0/1/0/0/2/3	100
HundredGigEctr0/1/0/2/4	ODU-FLEX0/1/0/0/2/4	100
HundredGigEctr0/1/0/3/1	ODU-FLEX0/1/0/0/3/1	100
HundredGigEctr0/1/0/3/2	ODU-FLEX0/1/0/0/3/2	100
HundredGigEctr0/1/0/3/3	ODU-FLEX0/1/0/0/3/3	100
HundredGigEctr0/1/0/3/4	ODU-FLEX0/1/0/0/3/4	100

The following is a sample to verify the 1200G 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:     1200G
Status:            Provisioned
LLDP Drop Enabled: FALSE
ARP Snoop Enabled: FALSE
Client Port        Mapper/Trunk Port        CoherentDSP0/1/0/7
Traffic Split Percentage

HundredGigEctr0/1/0/4/1  ODU-FLEX0/1/0/7/4/1    100
HundredGigEctr0/1/0/4/2  ODU-FLEX0/1/0/7/4/2    100
HundredGigEctr0/1/0/4/3  ODU-FLEX0/1/0/7/4/3    100
HundredGigEctr0/1/0/4/4  ODU-FLEX0/1/0/7/4/4    100
HundredGigEctr0/1/0/5/1  ODU-FLEX0/1/0/7/5/1    100
HundredGigEctr0/1/0/5/2  ODU-FLEX0/1/0/7/5/2    100
HundredGigEctr0/1/0/5/3  ODU-FLEX0/1/0/7/5/3    100
HundredGigEctr0/1/0/5/4  ODU-FLEX0/1/0/7/5/4    100
HundredGigEctr0/1/0/6/1  ODU-FLEX0/1/0/7/6/1    100
HundredGigEctr0/1/0/6/2  ODU-FLEX0/1/0/7/6/2    100
HundredGigEctr0/1/0/6/3  ODU-FLEX0/1/0/7/6/3    100
HundredGigEctr0/1/0/6/4  ODU-FLEX0/1/0/7/6/4    100
```

Muxponder Mode Configuration on the 2.4TX Card

Table 7: Feature History

Feature Name	Release Information	Feature Description
Bandwidth Splitting for Efficient Trunk Port Utilization	Cisco IOS XR Release 24.1.1	The NCS1K14-2.4T-X-K9 card can split traffic on a client port between the two trunk ports. This feature is useful when you configure the line card for a trunk capacity of 600GE (supplied by two 400GE client ports) or 1000GE (supplied by three 400GE client ports), which leaves 200GE surplus data on one client port. In the Muxponder mode, the other trunk port consumes this 200GE surplus data from a specific shared client port, effectively splitting the 400GE client port capacity equally across two trunks. The shared client ports are Port 2 for 600G and Port 3 for 1000G trunk payloads.

The muxponder mode enables the 2.4TX card to split wavelengths in specific client ports between the two trunk ports. In the slice mode, the client ports that support wavelength splitting act the same as other client ports. However, in the muxponder mode, the 2.4TX card activates the split client ports. The shared client ports are client port 2 for 600G and client port 3 for 1000G.

For example, if you set the data rate as 600G per trunk, client ports 1 and 4 carry 400GE each and the shared port 2 equally splits the 400GE traffic between the Trunk ports. Hence the Trunk Ports 0 and 7 always carry the same trunk rates in the muxponder mode.



Note In muxponder mode,

- In a point-to-point topology, port 0 in the near end node must be connected to port 0 in the far end node. The same is applicable to port 7 in the near end and port 7 in the far end nodes.
- In a point-to-point topology, the length of the optic fibers that are connected to both the trunk ports (0 and 7) must be the same. The fiber length difference must be less than 500 m. If the fiber length difference is more than 500 m, then the traffic on the split port does not come up.

Example Scenario

For 600G trunk rate, you must configure Port 1 as 400GE, Port 2 as 400GE, and Ports 4 as 400GE. Trunk 0 receives 400GE from port 1. Trunk 7 receives 400GE from port 4. As per split client configuration, Port 2 gives 200GE to Trunk 0 and another 200GE to Trunk 7. In this way, both trunk ports carry 600G data rate each.

Commands

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

configuration

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

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

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

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

commit

Client Rate and Pluggable Combinations

The following table shows the client rate and client pluggable combinations that are supported for the muxponder mode trunk rates.

Table 8: Trunk Rate and Client Pluggable Combinations

Trunk Rate	Client Rate Combination	Client Pluggable Combination	Client Ports	Shared Client Port
600G	Configure the 600G Trunk in the Muxponder Mode	3x QDD-400G	1, 4	2
1000G	Configure the 1000G Trunk in the Muxponder Mode	5x QDD-400G ⁵	1, 2, 4, 5	3

⁵ QDD-400G refers to QDD-400G-FR4-S, QDD-400G-AOCxM, and QDD-400G-DR4-S pluggables.

Configure the Muxponder Mode on the 2.4TX Card

The following sections describe the configurations for each trunk rate in the muxponder mode.

Configure the 600G Trunk in the Muxponder Mode

Commands

To configure the 2.4TX card in the 600G muxponder mode, use the following commands:

configuration

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

trunk-rate 600G

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

commit

Configuration

The following is a sample in which the 2.4TX card is configured in the *mxponder* mode for 600G trunk rate. This configuration has 400GE client on ports 1, 2, and 4.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/2/NXR0
RP/0/RP0/CPU0:ios(config-hwmod)#mxponder
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 client-type 400GE
```

```
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 600G Muxponder Mode

The following is a sample to verify the configured 600G muxponder mode. In the following output, the 400GE client traffic in *port 2 is split* between the two trunk coherentDSP controllers.

```
RP/0/RP0/CPU0:ios#show hw-module location 0/2/NXR0 mxponder

Location:                0/2/NXR0
Client Bitrate:          400GE
Trunk Bitrate:           600G
Status:                   Provisioned
LLDP Drop Enabled:       FALSE
ARP Snoop Enabled:       FALSE
Client Port              Mapper/Trunk Port          CoherentDSP0/2/0/0
CoherentDSP0/2/0/7      Traffic Split Percentage

FourHundredGigECtrlr0/2/0/1  ODU-FLEX0/2/0/0/1          100
0
FourHundredGigECtrlr0/2/0/2  ODU-FLEX0/2/0/0/2          50
50
FourHundredGigECtrlr0/2/0/4  ODU-FLEX0/2/0/7/4          0
100
```

Configure the 1000G Trunk in the Muxponder Mode

Commands

To configure the 2.4TX card in the 1000G muxponder mode, use the following commands:

configuration

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

trunk-rate 1000G

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

commit

Configuration

The following is a sample in which the 2.4TX card is configured in the *mxponder* mode for 1000G trunk rate. This configuration has 400GE client on ports 1, 2, 3, 4, and 5.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#hw-module location 0/2/NXR0
RP/0/RP0/CPU0:ios(config-hwmod)#mxponder
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 client-type 400GE
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 1000G Muxponder Mode

The following is a sample to verify the configured 1000G muxponder mode. In the following output, the 400GE client traffic in *port 3 is split* between the two trunk coherentDSP controllers.

```
RP/0/RP0/CPU0:ios#show hw-module location 0/2/NXR0 muxponder

Location:                0/2/NXR0
Client Bitrate:          400GE
Trunk Bitrate:           1000G
Status:                  Provisioned
LLDP Drop Enabled:       FALSE
ARP Snoop Enabled:       FALSE
Client Port              Mapper/Trunk Port          CoherentDSP0/2/0/0
CoherentDSP0/2/0/7

                                Traffic Split Percentage

FourHundredGigEctrlr0/2/0/1    ODU-FLEX0/2/0/0/1                100
0
FourHundredGigEctrlr0/2/0/2    ODU-FLEX0/2/0/0/2                100
0
FourHundredGigEctrlr0/2/0/3    ODU-FLEX0/2/0/0/3                50
50
FourHundredGigEctrlr0/2/0/4    ODU-FLEX0/2/0/7/4                0
100
FourHundredGigEctrlr0/2/0/5    ODU-FLEX0/2/0/7/5                0
100
```

QXP Card

Table 9: Feature History

Feature Name	Release Information	Description
NCS1K4-QXP-K9 Line Card Support on NCS 1014	Cisco IOS XR Release 24.1.1	NCS1K4-QXP-K9 line card delivers low cost 100G and 400G DWDM transmission with ZR+ optics on a router. This line card can be used in both traditional Optical Networking solution and in Routed Optical Networking solution. This line card has 16 pluggable ports with eight QSFP-DD client ports and eight QSFP-DD trunk ports. For more information about the NCS1K4-QXP-K9 card, see the datasheet .

The NCS1K4-QXP-K9 3.2T QSFP-DD DCO Transponder Line Card has eight client ports (QSFP-DD) and eight trunk ports (QSFP-DD ZR+). Each line card supports up to 3.2 Tbps traffic. The client rates that are supported are 400GE, 4x100GE, and 100GE Ethernet only. The modulation formats supported are 16 QAM for 400GE Txp/4x100GE Mxp.

The QXP line card provides up to 16 QSFP-DD ports (eight QSFP-DD client ports and eight QSFP-DD trunk ports). The supported operating modes are:

- 400GE-TXP
- 4X100GE MXP
- 2x100GE MXP

The QXP card has 8 slices. Each slice consists of one client and one trunk port with a slice capacity of 400G. The total capacity is 3.2T.

Table 10: Slice and Port Mapping on the QXP Card

Slice	Trunk Port	Client Port
0	0	1
1	2	3
2	4	5
3	6	7
4	8	9
5	10	11
6	12	13
7	14	15



Note When you use OPENROADM trunk mode by configuring the **trunk-mode OR** command, use only alternate slices on the QXP card. Either use slices 0, 2, 4, 6 or 1, 3, 5, 7.

Supported Data Rates for QXP Card

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

Operating mode	Card Support	Client Data Rate	Client Optics	Trunk Ports	Client Ports
400GE-TXP	QXP Card	400G	QDD-400G-DR4-S, QDD-400G-FR4-S, QDD-AOCxM	0,2,4,6,8,10,12,14	1,3,5,7,9,11,13,15
4X100GE MXP	QXP Card	4X100G Break out	QDD-400G-DR4-S, QDD-4X100G-LR-S	0,2,4,6,8,10,12,14	1,3,5,7,9,11,13,15
2X100GE MXP	QXP Card	2X100G Break out	QDD-400G-DR4-S, QDD-4X100G-LR-S	0,2,4,6,8,10,12,14	1,3,5,7,9,11,13,15

Configure 400G Transponder Mode

Use the following commands to configure and provision 400G TXP.

hw-module location *location*

mxponder-slice *slice-number*

trunk-rate 400G

trunk-mode [ZR | OR]

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

The following is a sample configuration of configuring a 400G TXP.

```
RP/0/RP0/CPU0:ios#configure
Tue Apr 11 19:29:20.132 UTC
RP/0/RP0/CPU0:ios(config)#hw-module location 0/1 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
```

The following is a sample output of **show hw-module location** *location* **mxponder-slice** *slice-number* when configured in 400G Transponder Mode.

```
RP/0/RP0/CPU0:ios#sh hw-module location 0/0 mxponder-slice 0
Sat Jun 25 21:32:58.799 UTC

Location:                0/0
Slice ID:                 0
Client Bitrate:          400GE
Trunk Bitrate:           400G
Status:                   Provisioned
LLDP Drop Enabled:       FALSE
ARP Snoop Enabled:       FALSE
Client Port               Mapper/Trunk Port      CoherentDSP0/0/0/0
                          Traffic Split Percentage
FourHundredGigEctrlr0/0/0/1      -                100
```



Note The **trunk-mode** command allows you to choose between OTN and ethernet traffic on the trunk port.

Configure 400G Muxponder Mode

Use the following commands to configure and provision 400G MXP.

hw-module location *location*

mxponder-slice *slice-number*

trunk-rate 400G

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

The following is a sample configuration of configuring a 400G MXP.

```
RP/0/RP0/CPU0:ios#configure
Tue Apr 11 19:29:20.132 UTC
RP/0/RP0/CPU0:ios(config)#hw-module location 0/0 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
```

The following is a sample output of **show hw-module location *location* mxponder-slice *slice-number*** when configured in 400G MXP Mode.

```
RP/0/RP0/CPU0:ios#sh hw-module location 0/3 mxponder-slice 1
Sat Jun 25 23:03:20.823 UTC

Location:                0/3
Slice ID:                 1
Client Bitrate:          100GE
Trunk Bitrate:           400G
Status:                  Provisioned
LLDP Drop Enabled:       FALSE
ARP Snoop Enabled:       FALSE
Client Port              Mapper/Trunk Port          CoherentDSP0/3/0/2
                          Traffic Split Percentage

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

Configure 2x100G Muxponder Mode

Use the following commands to configure and provision 2x100G MXP.

hw-module location *location*

mxponder-slice *slice-number*

trunk-rate 200G

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

The following is a sample configuration of configuring a 2x100G MXP.

```
RP/0/RP0/CPU0:ios#configure
Tue Apr 11 19:29:20.132 UTC
RP/0/RP0/CPU0:ios(config)#hw-module location 0/0 mxponder-slice 0
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 200G
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
```

The following is a sample output of **show hw-module location *location* mxponder-slice *slice-number*** when configured in 2x100G MXP Mode.

```
RP/0/RP0/CPU0:ios#sh hw-module location 0/3 mxponder-slice 1
Sat Jun 25 23:03:20.823 UTC

Location:                0/3
Slice ID:                 1
Client Bitrate:          100GE
Trunk Bitrate:           200G
Status:                  Provisioned
LLDP Drop Enabled:       FALSE
ARP Snoop Enabled:       FALSE
Client Port              Mapper/Trunk Port          CoherentDSP0/3/0/2
                          Traffic Split Percentage

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

DAC Supported Modes for NCS1K4-QXP-K9 Card

DAC support is enabled on the NCS1K4-QXP-K9 card for 2x100G, 4x100G, and 400G operating modes. The following table provides the details of the respective DAC rates for the different trunk rates for NCS1K4-QXP-K9 card.

Table 11: DAC Supported Data Rates for NCS1K4-QXP-K9 Card

Trunk Rate	Modulation Format	Default Value	Modified DAC Supported
200G	QPSK	1x1	1x1.50
200G	8QAM	1x1.25	N/A
200G	16-QAM	1x1.25	N/A
400G	16-QAM	1x1	1x1.50

The following example changes the DAC rate to 1x1.5 on an optics controller.

```
RP/0/RP0/CPU0:ios(config)#controller optics 0/0/0/0
RP/0/RP0/CPU0:ios(config-Optics)#dac-Rate 1x1.50
RP/0/RP0/CPU0:ios(config-Optics)#commit
```



- Note**
- Changing the DAC turns the laser Off and then back on for the optics. This is a traffic impacting operation.
 - The DAC rate configuration must match on both ends of a connection.

Cisco 400G QSFP-DD High-Power (Bright ZR+) Optical Module Support on QXP Card

QXP card supports Cisco 400G QSFP-DD High-Power (Bright) Optical Modules. DP04QSDD-HK9 operates as Ethernet or OTN transponder. DP04QSDD-HE0 operates only as an Ethernet transponder.

Use the following commands to configure OTN data path on the Bright ZR+ pluggable optical modules. The **trunk-mode OR** refers to OpenROADM.

hw-module location *location*

mxponder-slice 1 *slice-number*

trunk-mode OR

trunk-rate *rate*

Use the following commands to configure Ethernet data path on the Bright ZR+ pluggable optical modules.

hw-module location *location*

mxponder-slice 1 *slice-number*

trunk-mode ZR

trunk-rate *rate*



Note DP04QSDD-HK9 operates as Ethernet or OTN transponder. DP04QSDD-HE0 operates only as an Ethernet transponder. DP04QSDD-HE0 supports only trunk-mode ZR. Configuring trunk-mode OR on the DP04QSDD-HE0 pluggable raises the MEA alarm.

The following is a sample configuration of configuring a 4x100G OTN trunk on a Bright ZR+ pluggable.

```
RP/0/RP0/CPU0:ios#configure
Tue Apr 11 19:29:20.132 UTC
RP/0/RP0/CPU0:ios(config)#hw-module location 0/0
RP/0/RP0/CPU0:ios(config-hwmod)#mxponder-slice 4
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-mode OR
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 400G
RP/0/RP0/CPU0:ios(config-hwmod-mxp)# client-port-rate 9 lane 1 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)# client-port-rate 9 lane 2 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)# client-port-rate 9 lane 3 client-type 100GE
RP/0/RP0/CPU0:ios(config-hwmod-mxp)# client-port-rate 9 lane 4 client-type 100GE
```

The following is a sample configuration of configuring Ethernet trunk on a Bright ZR+ pluggable.

```
RP/0/RP0/CPU0:ios#configure
Tue Apr 11 19:29:20.132 UTC
RP/0/RP0/CPU0:ios(config)#hw-module location 0/0
RP/0/RP0/CPU0:ios(config-hwmod)#mxponder-slice 4
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-mode ZR
RP/0/RP0/CPU0:ios(config-hwmod-mxp)#trunk-rate 400G
```

The following is a sample configuration of setting 0dBm transmit power on a Bright ZR+ pluggable.

```
RP/0/RP0/CPU0:ios#configure
RP/0/RP0/CPU0:ios(config)#controller optics 0/0/0/2
RP/0/RP0/CPU0:ios(config-Optics)#transmit-power 0
Thu Mar 9 13:02:30.662 UTC
WARNING! Changing TX power can impact traffic
RP/0/RP0/CPU0:ios(config-Optics)#commit
Thu Mar 9 13:02:31.566 UTC
```

The following is a sample output of the **show controllers optics** command, with the transmit power set to 0 dBm.

```
RP/0/RP0/CPU0:ios#show controllers optics 0/0/0/8
Thu Apr 13 13:54:33.163 UTC
Controller State: Up
Transport Admin State: In Service
Laser State: On
LED State: Green
Optics Status
  Optics Type: QSFP-DD DWDM
  DWDM carrier Info: C BAND, MSA ITU Channel=49, Frequency=193.70THz,
  Wavelength=1547.715nm
  Alarm Status:
  -----
  Detected Alarms: None
  LOS/LOL/Fault Status:
  Alarm Statistics:
  -----
  HIGH-RX-PWR = 0          LOW-RX-PWR = 4
  HIGH-TX-PWR = 0          LOW-TX-PWR = 1
  HIGH-LBC = 0            HIGH-DGD = 0
  OOR-CD = 0              OSNR = 4
  WV-L-OOL = 0            MEA = 0
  IMPROPER-REM = 0
  TX-POWER-PROV-MISMATCH = 0
```


Laser Bias Current = 0.0 %
 Actual TX Power = 0.00 dBm
 RX Power = -10.50 dBm
 RX Signal Power = -10.35 dBm
 Frequency Offset = 199 MHz

Performance Monitoring: Enable

THRESHOLD VALUES

Parameter	High Alarm	Low Alarm	High Warning	Low Warning
Rx Power Threshold(dBm)	3.0	-24.5	0.0	0.0
Tx Power Threshold(dBm)	0.0	-16.0	0.0	0.0
LBC Threshold(mA)	N/A	N/A	0.00	0.00

LBC High Threshold = 90 %
 Configured Tx Power = 0.00 dBm
 Configured CD High Threshold = 52000 ps/nm
 Configured CD lower Threshold = -52000 ps/nm
 Configured OSNR lower Threshold = 21.10 dB
 Configured DGD Higher Threshold = 67.00 ps

Table 12: Operating Modes Supported for Bright ZR+ Pluggable Modules on QXP Card

Operating mode	Modulation	FEC
4x100GE MXP	16-QAM	CFEC
4x100GE MXP	16-QAM	OFEC
2x100GE MXP	QPSK	OFEC
400GE TXP	16-QAM	CFEC
400GE TXP	16-QAM	OFEC

