Configuring DSPU and SNA Service Point Support

This chapter describes our support for Systems Network Architecture (SNA) downstream physical unit (DSPU) devices and SNA Service Point. For a complete description of the commands mentioned in this chapter, refer to the "DSPU and SNA Service Point Configuration Commands" chapter of the *Bridging and IBM Networking Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

DSPU Configuration Task List

To configure DSPU, perform the tasks in the following sections. The last two tasks are optional.

- Define DSPU Upstream Hosts
- Define Downstream PUs
- Define DSPU LUs
- Configure DSPU to Use a Data Link Control
- Define the Number of Outstanding, Unacknowledged Activation RUs
- Configure SNA Service Point Support
- Monitor DSPU and SNA Service Point Feature Status

See the end of this chapter for "DSPU and SNA Service Point Configuration Examples."

Define DSPU Upstream Hosts

The upstream host provides logical units (LUs) that the Cisco IOS software assigns for use by its downstream PUs. Because one upstream host can only provide a maximum of 255 LUs, the DSPU feature supports multiple hosts. Multiple upstream host support allows the DSPU router to provide more than 255 LUs for use by its downstream PUs.

To define a DSPU host over Token Ring, Ethernet, FDDI, remote source-route bridging (RSRB), or virtual data-link control connections, perform the following task in global configuration mode:

Task	Command
Define a DSPU host over Token Ring, Ethernet, FDDI, RSRB, or virtual data-link control connections.	**dspu host** *host-name* **xid-snd** *xid* **rmac** *remote-mac* [**rsap** *remote-sap*] [**lsap** *local-sap*] [**interface** *slot/port*] [**window** *window-size*] [**maxiframe** *max-iframe*] [**retries** *retry-count*] [**retry-timeout** *retry-timeout*] [**focalpoint**]

To define a DSPU host over an SDLC connection, perform the following task in global configuration mode:

Task	Command
Define a DSPU host over an SDLC connection	**dspu host** *host-name* **xid-snd** *xid* **sdlc** *sdlc-addr* [**interface** *slot/port*] [**window** *window-size*] [**maxiframe** *max-iframe*] [**retries** *retry-count*] [**retry-timeout** *retry-timeout*] [**focalpoint**]

To define a DSPU host over an X.25/QLLC connection, perform the following task in global configuration mode:

Task	Command
Define a DSPU host over an X.25/QLLC connection.	**dspu host** *host-name* **xid-snd** *xid* **x25** *remote-x121-addr* [**qllc** *local-x121-subaddr*] [**interface** *slot/port*] [**window** *window-size*] [**maxiframe** *max-iframe*] [**retries** *retry-count*] [**retry-timeout** *retry-timeout*] [**focalpoint**]

To define a DSPU host over a Frame Relay connection, perform the following task in global configuration mode:

Task	Command
Define a DSPU host over a Frame Relay connection.	**dspu host** *host-name* **xid-snd** *xid* **dlci** *dlci-number* [**rsap** *remote-sap*] [**lsap** *local-sap*] [**interface** *slot/port*] [**window** *window-size*] [**maxiframe** *max-iframe*] [**retries** *retry-count*] [**retry-timeout** *retry-timeout*] [**focalpoint**]

Define Downstream PUs

To define the downstream PUs, perform either of the tasks in the following sections, depending on your circumstances:

- Explicitly Define a Downstream PU
- Enable the Default PU Option

Explicitly Define a Downstream PU

Explicitly define a downstream PU if you require the Cisco IOS software to perform verification checking on incoming downstream connections or to initiate an outgoing downstream connection.

For Cisco IOS 11.3 and above, the number of DSPU PUs you can configure is 1024.

To explicitly define a downstream PU over Token Ring, Ethernet, FDDI, RSRB, virtual data-link control, or NCIA connections, perform the following task in global configuration mode:

Task	Command
Explicitly define a downstream PU over Token Ring, Ethernet, FDDI, RSRB, virtual data-link control, or NCIA connections.	**dspu pu** *pu-name* [**rmac** *remote-mac*] [**rsap** *remote-sap*] [**lsap** *local-sap*] [**xid-rcv** *xid*] [**interface** *slot/port*] [**window** *window-size*] [**maxiframe** *max-iframe*] [**retries** *retry-count*] [**retry-timeout** *retry-timeout*]

To explicitly define a downstream PU over an SDLC connection, perform the following task in global configuration mode:

Task	Command
Explicitly define a downstream PU over an SDLC connection.	**dspu pu** *pu-name* [**sdlc** *sdlc-addr*] [**xid-rcv** *xid*] [**interface** *slot/port*] [**window** *window-size*] [**maxiframe** *max-iframe*] [**retries** *retry-count*] [**retry-timeout** *retry-timeout*]

To explicitly define a downstream PU over an X.25/QLLC connection, perform the following task in global configuration mode:

Task	Command
Explicitly define a downstream PU over an X.25/QLLC connection.	**dspu pu** *pu-name* [**x25** *remote-x121-addr*] [**qllc** *local-x121-subaddr*] [**xid-rcv** *xid*] [**interface** *slot/port*] [**window** *window-size*] [**maxiframe** *max-iframe*] [**retries** *retry-count*] [**retry-timeout** *retry-timeout*]

To explicitly define a downstream PU over a Frame Relay connection, perform the following task in global configuration mode:

Task	Command
Explicitly define a downstream PU over a Frame Relay connection.	**dspu pu** *pu-name* [**dlci** *dlci-number*] [**rsap** *remote-sap*] [**lsap** *local-sap*] [**xid-rcv** *xid*] [**interface** *type slot/port*] [**window** *window-size*] [**maxiframe** *max-iframe*] [**retries** *retry-count*] [**retry-timeout** *retry-timeout*]

Note that a PU definition must have either an xid-rcv parameter or an address (rmac, sdlc, x25 or dlci) parameter.

If the Cisco IOS software will perform verification checking on incoming downstream connections, there are several combinations of parameters that you can configure for verification matching. Note that the address parameter, when specified, is considered to be the primary key on the PU definition. Therefore, if both an address and xid-rcv are configured, the matching algorithm will match on the address and ignore the xid-rcv parameter.

- Match on xid-rcv value only

  User may define a downstream PU using only the xid-rcv value so that any connecting PU that specifies the value of the configured XID will match that PU definition.

- Match on xid-rcv and interface values

  User may define a downstream PU using the xid-rcv and interface values so that any PU connecting into the configured interface that specifies the value of the configured XID will match the PU definition.

- Match on addressing values only

  User may define a downstream PU using only the addressing values (RMAC/RSAP/LSAP, SDLC, DLCI/RSAP/LSAP, or X25/QLLC) so that any connecting PU with addressing that matches the configured addressing will match that PU definition. If no PU definition is found to match the incoming RSAP, then a match is accepted on a PU that has the correct RMAC/LSAP or DLCI/LSAP.

- Match on addressing and interface values

  User may define a downstream PU using the interface and addressing values (RMAC/RSAP/LSAP, SDLC, DLCI/RSAP/LSAP, or X25/QLLC) so that any PU connecting into the configured interface with addressing that matches the configured addressing will match the PU definition. If no PU definition is found to match the incoming RSAP, then a match is accepted on a PU that has the correct RMAC/LSAP or DLCI/LSAP and interface.

The Cisco IOS software rejects any incoming downstream connections that do not match the parameters of a defined downstream PU unless the default PU option is also enabled.

Enable the Default PU Option

Configure the DSPU default PU option if you do not require the Cisco IOS software to verify incoming downstream connections. The default PU option allows the software to accept incoming downstream connections without an explicit definition for the downstream PU.

To enable the default PU option, perform the following task in global configuration mode:

Task	Command
Enable the default PU option.	**dspu default-pu** [**window** *window-size*] [**maxiframe** *max-iframe*]

Define DSPU LUs

Specify the LU routing algorithm used to map the upstream LUs to the downstream LUs and to define all LUs for each upstream and downstream PU.

The DSPU feature assigns upstream LUs to downstream LUs based on the selected LU routing algorithm and performs the mapping necessary for SNA data transfer.

The DSPU feature supports two alternative mapping algorithms that are described in the following sections:

- Define Dedicated LU Routing
- Define Pooled LU Routing

An upstream host PU or downstream PU can support up to 255 LU sessions. The DSPU feature allows each LU to be individually configured for either dedicated LU routing or pooled LU routing.

Define Dedicated LU Routing

You can configure an upstream LU so that it is reserved, or dedicated, for use by a specific downstream LU.

To define a dedicated LU or a range of dedicated LUs for an upstream host and downstream PU, perform the following task in global configuration mode:

Task	Command
Define a dedicated LU or a range of dedicated LUs for a downstream PU.	**dspu lu** *lu-start* [*lu-end*] {**host** *host-name host-lu-start* | **pool** *pool-name*} [**pu** *pu-name*]

See the "Dedicated LU Routing Example" section later in this chapter for an example of dedicated LU routing.

Define Pooled LU Routing

You can configure an upstream host LU so that it is a member of a pool of LUs. When a downstream connection is established and the downstream LU is configured as a pooled LU, the Cisco IOS software selects an upstream LU from the pool for assignment to the downstream LU.

Pooled LU routing allows a limited number of upstream host LUs to be shared (at different times) among many downstream LUs.

To define a host LU or a range of host LUs in an LU pool, perform the following task in global configuration mode:

Task	Command
Define a host LU or a range of host LUs in an LU pool.	**dspu pool** *pool-name* **host** *host-name* **lu** *lu-start* [*lu-end*] [**inactivity-timeout** *inactivity-minutes*]

You can configure a downstream LU as a pooled LU. When a downstream connection is established and the downstream LU is configured as a pooled LU, the software selects an upstream LU from the specified pool for assignment to the downstream LU.

To define a pooled LU or a range of pooled LUs for a downstream PU, perform the following task in global configuration mode:

Task	Command
Define a pooled LU or a range of pooled LUs for a downstream PU.	**dspu lu** *lu-start* [*lu-end*] **pool** *pool-name* **pu** *pu-name*

See the "Pooled LU Routing Example" section later in this chapter for an example of pooled LU routing.

Configure DSPU to Use a Data Link Control

The final step in configuring DSPU is to define the data link controls that will be used for upstream host and downstream PU connections.

The DSPU feature supports the data link controls described in the following sections:

- Configure DSPU to Use Token Ring, Ethernet, or FDDI
- Configure DSPU to Use RSRB

- Configure DSPU to Use RSRB with Local Acknowledgment
- Configure DSPU to Use Virtual Data-link Control
- Configure DSPU to Use SDLC
- Configure DSPU to Use QLLC
- Configure DSPU to Use Frame Relay
- Configure DSPU to Use NCIA

Configure DSPU to Use Token Ring, Ethernet, or FDDI

You can configure DSPU to use the Token Ring, Ethernet, or FDDI data link controls or both by enabling a service access point (SAP) address on the interface. Each interface can support up to 254 local SAPs enabled for either upstream or downstream connections; a local SAP cannot be enabled for both upstream and downstream connections on the same interface.

To enable a local SAP on the Token Ring, Ethernet, or FDDI interfaces for use by upstream hosts, perform the following task in interface configuration mode:

Task	Command
Enable local SAP for upstream hosts.	**dspu enable-host** [**lsap** *local-sap*]

To enable a local SAP on the Token Ring, Ethernet, or FDDI interfaces for use by downstream PUs, perform the following task in interface configuration mode:

Task	Command
Enable local SAP for downstream PUs.	**dspu enable-pu** [**lsap** *local-sap*]

Once a local SAP is enabled, it is ready to accept incoming connection attempts from the remote device (upstream host or downstream PU). Alternatively, initiate an outgoing connection to the remote device by performing the following task in interface configuration mode:

Task	Command
Initiate connection with upstream host or downstream PU via Token Ring or Ethernet.	**dspu start** {*host-name* | *pu-name*}

Configure DSPU to Use RSRB

To configure DSPU to use RSRB, you must create a DSPU/RSRB data link control.

Similar to our implementation of SDLLC, the DSPU/RSRB data link control uses the concept of a virtual Token Ring device residing on a virtual Token Ring to represent the Cisco IOS software to upstream hosts and downstream PUs across an RSRB network.

Because the upstream host and downstream PU expects its peer to also be on a Token Ring, you must assign a virtual Token Ring address (the DSPU virtual MAC address) to the DSPU/RSRB data link control. Like real Token Ring addresses, the DSPU virtual MAC address must be unique across the network.

In addition to assigning the DSPU virtual MAC address, you must also assign a DSPU virtual ring number to the DSPU/RSRB data link control. The DSPU virtual ring number must be unique across the network.

Note The DSPU virtual ring number is a different number from the virtual ring group numbers that you use to configure RSRB and multiport bridging.

The combination of the DSPU virtual MAC address and the DSPU virtual ring number identifies the DSPU/RSRB data link control interface to the rest of an RSRB network.

When an end station (either an upstream host or a downstream PU) attempts to connect with the DSPU software, the following events occur:

1. The end station sends explorer packets with the locally administered MAC address on the router interface to which the end station is connected.
2. The router configured with that locally administered MAC address or with the hardware MAC address intercepts the frame, fills in the DSPU virtual ring number and the DSPU bridge number in the routing information field (RIF), and sends a response to the end station.
3. The end station establishes a session with the DSPU router.

To define the DSPU/RSRB data link control interface, perform the following tasks in global configuration mode:

Task	Command
Define an RSRB ring group.	**source-bridge ring-group** *ring-group* [*virtual-mac-address*]
Define a remote peer with the local acknowledgment feature.	**source-bridge remote-peer** *ring-group* **tcp** *ip-address* **local-ack**
Define the DSPU/RSRB interface.	**dspu rsrb** *local-virtual-ring bridge-number target-virtual-ring virtual-macaddr*

After you define the DSPU RSRB data link control, configure DSPU to use the RSRB data link control by enabling a local SAP for either upstream or downstream connections.

To enable a local SAP on RSRB for use by upstream hosts, perform the following task in global configuration mode:

Task	Command
Enable local SAP for upstream hosts.	**dspu rsrb enable-host** [**lsap** *local-sap*]

To enable a local SAP on RSRB for use by downstream PUs, perform the following task in global configuration mode:

Task	Command
Enable local SAP for downstream PUs.	**dspu rsrb enable-pu** [**lsap** *local-sap*]

Once a local SAP is enabled, it is ready to accept incoming connection attempts from the remote device (upstream host or downstream PU) over RSRB. Alternatively, initiate an outgoing connection to the remote device by performing the following task in global configuration mode:

Task	Command
Initiate connection with upstream host or downstream PU via RSRB.	**dspu rsrb start** {*host-name* | *pu-name*}

Configure DSPU to Use RSRB with Local Acknowledgment

Configuring DSPU to use RSRB with local acknowledgment is identical to configuring RSRB with local acknowledgment. If you add the **local-ack** keyword to the **source-bridge remote-peer** configuration command, DSPU will use local acknowledgment for any end stations that connect to DSPU from that peer.

To configure DSPU to use RSRB with local acknowledgment, perform the following tasks in global configuration mode:

Task	Command
Define an RSRB ring group.	**source-bridge ring-group** *ring-group* [*virtual-mac-address*]
Define a remote peer with the local acknowledgment feature.	**source-bridge remote-peer** *ring-group* **tcp** *ip-address* **local-ack**
Define the DSPU/RSRB interface.	**dspu rsrb** *local-virtual-ring bridge-number target-virtual-ring virtual-macaddr*

Configure DSPU to Use Virtual Data-link Control

To configure DSPU to use virtual data-link control, you must create a DSPU virtual data-link control interface.

Similar to our implementation of SDLLC, the DSPU virtual data-link control interface uses the concept of a virtual Token Ring device residing on a virtual Token Ring to represent the Cisco IOS software to upstream hosts and downstream PUs across a network.

Because the upstream host and downstream PU expects its peer to also be on a Token Ring, you must assign a virtual Token Ring address (the DSPU virtual MAC address) to the DSPU virtual data-link control interface. Like real Token Ring addresses, the DSPU virtual MAC address must be unique across the network.

In addition to assigning the DSPU virtual MAC address, you must also identify the source-route bridging virtual ring number with which the DSPU virtual MAC address will be associated. The source-route bridging virtual ring number is set using the **source-bridge ring-group** command, which is documented in the "Source-Route Bridging Commands" chapter of the *Bridging and IBM Networking Command Reference*.

The combination of the DSPU virtual MAC address and the source-route bridging virtual ring number identifies the DSPU virtual data-link control interface to the rest of the DLSw+ network.

When an end station (either an upstream host or a downstream PU) attempts to connect with the DSPU software, the following events occur:

1. The end station sends explorer packets with the locally administered MAC address on the router interface to which the end station is connected.
2. The router configured with that locally administered MAC address intercepts the frame, DLSw+ adjusts the routing information field (RIF), and sends a response to the end station.
3. The end station establishes a session with the DSPU router.

Prior to creating the DSPU virtual data-link control interface, you must also configure DLSw+ peers so that DLSw+ can provide the communication path. The commands for defining DLSw+ local and remote peers are documented in the "DLSw+ Configuration Commands" chapter of the *Bridging and IBM Networking Command Reference*.

To define the DSPU virtual data-link control interface, perform the following task in global configuration mode:

Task	Command
Define the DSPU virtual data-link control interface.	**dspu vdlc** *ring-group virtual-mac-address*

After you define the DSPU virtual data-link control interface, configure DSPU to use virtual data-link control by enabling a local SAP for either upstream or downstream connections.

To enable a local SAP on the virtual data-link control for use by upstream hosts, perform the following task in global configuration mode:

Task	Command
Enable local SAP for upstream hosts.	**dspu vdlc enable-host** [**lsap** *local-sap*]

To enable a local SAP on the virtual data-link control for use by downstream PUs, perform the following task in global configuration mode:

Task	Command
Enable local SAP for downstream PUs.	**dspu vdlc enable-pu** [**lsap** *local-sap*]

Once a local SAP is enabled, it is ready to accept incoming connection attempts from the remote device (upstream host or downstream PU) using virtual data-link control. Alternatively, initiate an outgoing connection to the remote device by performing the following task in global configuration mode:

Task	Command
Initiate connection with upstream host or downstream PU via virtual data-link control.	**dspu vdlc start** {*host-name* | *pu-name*}

Configure DSPU to Use SDLC

Before DSPU may be configured to use the SDLC data link control, the serial interface must be defined for SDLC encapsulation and assigned an SDLC role.

To define the serial interface to use SDLC and specify the SDLC role, perform the following tasks in interface configuration mode:

Task	Command
Enable SDLC encapsulation on the serial interface.	**encapsulation sdlc**
Specify the SDLC role of the router.	**sdlc role** {**none** | **primary** | **secondary** | **prim-xid-poll**}

For the connection to be established without XID exchange, the SDLC role must be **primary** if DSPU will be initiating connections to the SDLC partner. The SDLC role must be **secondary** or **none** if the SDLC partner will be initiating connections with DSPU.

When an XID exchange is required, the SDLC role must be **prim-xid-poll** or **none** if DSPU will be initiating connections to the SDLC partner. The role must be **none** if the SDLC partner will be initiating connections with DSPU.

The SDLC address(es) used on the SDLC link must also be defined. If DSPU is configured to initiate the connection, then the SDLC address identifies the SDLC partner. If the remote SDLC device initiates the connection, then the SDLC address identifies the address for which a connection will be accepted.

To configure the SDLC address, perform the following task in interface configuration mode:

Task	Command
Define the SDLC address.	**sdlc address** *hexbyte*

Finally, the SDLC address must be enabled for use by DSPU. Each interface can support up to 255 SDLC addresses enabled for either upstream or downstream connections; an SDLC address cannot be enabled for both upstream and downstream connections on the same interface. If the SDLC role is **none**, there can be only one SDLC address on that interface.

To enable an SDLC address for use by upstream host connections, perform the following task in interface configuration mode:

Task	Command
Enable SDLC address for upstream host.	**dspu enable-host sdlc** *sdlc-address*

To enable an SDLC address for use by downstream PU connections, perform the following task in interface configuration mode:

Task	Command
Enable SDLC address for downstream PU.	**dspu enable-pu sdlc** *sdlc-address*

When the SDLC role is configured as **primary**, DSPU initiates a connection with the remote device by sending set normal response mode (SNRM) when the SDLC address is enabled for DSPU.

When the SDLC role is configured as **prim-xid-poll**, DSPU initiates a connection with the remote device by sending a NULL XID when the SDLC address is enabled for DSPU.

When the SDLC role is configured as **secondary**, DSPU will not be ready to respond to SNRM until a **dspu start** *pu-name* command is issued.

When the SDLC role is configured as **none**, DSPU is ready to respond to a received XID or SNRM when the SDLC address is enabled for DSPU; otherwise, the connection may be initiated by issuing the **dspu start** *pu-name* command.

To configure DSPU to respond to SNRM when the SDLC role is configured as **secondary**, or to initiate a connection when the SDLC role is configured as **none**, perform the following task in interface configuration mode:

Task	Command
Initiate a connection with a remote device when the SDLC role is configured as **secondary** or **none**.	**dspu start** {*host-name* | *pu-name*}

Configure DSPU to Use QLLC

Before DSPU may be configured to use the QLLC data link control, the serial interface must be defined for X.25 encapsulation and assigned an X.121 address.

To define the serial interface to use X.25, perform the following tasks in interface configuration mode:

Task	Command
Enable X.25 encapsulation on serial interface.	**encapsulation x25** [**dce**]
Define an X.121 address.	**x25 address** *x121-addr*

X.25 routing must also be configured so that incoming calls to the local X.121 address can be appropriately routed to the serial interface and mapped into the QLLC data link control.

To define X.25 routing, perform the following tasks in global configuration mode:

Task	Command
Enable X.25 routing.	**x25 routing**
Enable routing of X.25 packets to serial interface.	**x25 route ^***local-x121-addr***.* alias serial slot/port**

To define which calls get mapped into QLLC, perform the following task in interface configuration mode:

Task	Command
Define remote X.121 address for mapping into QLLC.	**x25 map qllc** *x121-addr*

Finally, the local X.121 subaddress must be enabled for use by DSPU. An X.121 subaddress can be enabled for either upstream or downstream connections; an X.121 subaddress cannot be enabled for both upstream and downstream connections on the same interface.

To enable an X.121 subaddress for use by upstream host connections via QLLC, perform the following task in interface configuration mode:

Task	Command
Enable X.121 subaddress for upstream host.	**dspu enable-host qllc** *x121-subaddress*

To enable an X.121 subaddress for use by downstream PU connections via QLLC, perform the following task in interface configuration mode:

Task	Command
Enable X.121 subaddress for downstream PU.	**dspu enable-pu qllc** *x121-subaddress*

Once an X.121 subaddress is enabled, it is ready to accept incoming connection attempts from the remote device (upstream host or downstream PU) over QLLC. Alternatively, initiate an outgoing connection to the remote device by performing the following task in interface configuration mode:

Task	Command
Initiate connection with upstream host or downstream PU via QLLC.	**dspu start** {*host-name* | *pu-name*}

Configure DSPU to Use Frame Relay

Before DSPU may be configured to use the LLC2/Frame Relay data link control, the serial interface must be defined for Frame Relay encapsulation.

To define the serial interface for Frame Relay encapsulation, perform the following task in interface configuration mode:

Task	Command
Enable Frame Relay encapsulation on a serial interface.	**encapsulation frame-relay ietf**

The DLCI used on the Frame Relay link must be mapped into LLC2.

To configure the mapping of a DLCI into LLC2, perform the following task in interface configuration mode:

Task	Command
Configure DLCI mapping into LLC2.	**frame-relay map llc2** *dlci-number*

Finally, the local SAP address must be enabled for use by DSPU. A SAP address can be enabled for either upstream or downstream connections; a SAP address cannot be enabled for both upstream and downstream connections on the same interface.

To enable a local SAP on the LLC2/Frame Relay interface for use by upstream hosts, perform the following task in interface configuration mode:

Task	Command
Enable local SAP for upstream hosts.	**dspu enable-host** [**lsap** *local-sap*]

To enable a local SAP for the LLC2/Frame Relay interface for use by downstream PUs, perform the following task in interface configuration mode:

Task	Command
Enable local SAP for downstream PUs.	**dspu enable-pu** [**lsap** *local-sap*]

Once a local SAP is enabled, it is ready to accept incoming connection attempts from the remote device (upstream host or downstream PU) over Frame Relay. Alternatively, initiate an outgoing connection to the remote device by performing the following task in interface configuration mode:

Task	Command
Initiate connection with upstream host or downstream PU via LLC2/Frame Relay.	**dspu start** {*host-name* | *pu-name*}

Configure DSPU to Use NCIA

To configure DSPU to use NCIA, you must perform the following tasks:

- Configure the NCIA server as the underlying transport mechanism.
- Enable a local SAP on the NCIA server for use by downstream PUs.

To configure the NCIA server as the underlying transport mechanism, perform the following task in global configuration mode:

Task	Command
Configure the NCIA server as the underlying transport mechanism.	**dspu ncia** *server-number*

To enable a local SAP on the NCIA server for use by downstream PUs, perform the following task in global configuration mode:

Task	Command
Enable local SAP for downstream PUs.	**dspu ncia enable-pu** [**lsap** *local-sap*]

Define the Number of Outstanding, Unacknowledged Activation RUs

The DSPU feature allows you to define the number of activation request/response units (RUs) such as ACTLUs or DDDLU NMVTs that can be sent by the Cisco IOS software before waiting for responses from the remote PU.

The DSPU activation window provides pacing to avoid depleting the router buffer pool during PU activation. Increasing the window size allows more LUs to become active in a shorter amount of time (assuming the required buffers for activation RUs are available). Decreasing the window size limits the amount of buffers the DSPU may use during PU activation. Typically, you do not need to change the default window size.

To define the number of unacknowledged activation RUs that can be outstanding, perform the following task in global configuration mode:

Task	Command
Define the number of unacknowledged activation RUs.	**dspu activation-window** *window-size*

Configure SNA Service Point Support

Our implementation of SNA Service Point support includes support for three commands: Alerts, RUNCMD, and Vital Product Data support.

Alert support is provided as the Cisco IOS software sends unsolicited Alerts to NetView (or an equivalent network management application) at the host. This function occurs at the various router interfaces and protocol layers within the device.

RUNCMD support enables you to send commands to the router from the NetView console using the NetView RUNCMD facility, and the router sends the relevant replies back to the RUNCMD screen. Some commands, such as **telnet**, **rsh**, **rlogin**, and **tn3270**, are not supported.

Vital Product Data support allows you to request Vital Product Data from the NetView console. The router replies to NetView with the relevant information.

To configure SNA Service Point support, perform the tasks in the following sections:

- Define a Link to an SNA Host
- Configure Service Point Support to Use a Data Link Control
- Specify Names for All Attached LANs
- Specify the Physical Location of the Router

Note You do not need to perform the tasks in the next section if you have configured a DSPU host with the **focalpoint** parameter.

Define a Link to an SNA Host

To define a link to an SNA host over Token Ring, Ethernet, FDDI, RSRB, or virtual data-link control connections, perform the following task in global configuration mode:

Task	Command
Define a link to an SNA host over Token Ring, Ethernet, FDDI, RSRB, or virtual data-link control connections.	**sna host** *host-name* **xid-snd** *xid* **rmac** *remote-mac* [**rsap** *remote-sap*] [**lsap** *local-sap*] [**interface** *slot/port*] [**window** *window-size*] [**maxiframe** *max-iframe*] [**retries** *retry-count*] [**retry-timeout** *retry-timeout*] [**focalpoint**]

To define a link to an SNA host over an SDLC connection, perform the following task in global configuration mode:

Task	Command
Define a link to an SNA host over an SDLC connection.	**sna host** *host-name* **xid-snd** *xid* **sdlc** *sdlc-addr* [**interface** *slot/port*] [**window** *window-size*] [**maxiframe** *max-iframe*] [**retries** *retry-count*] [**retry-timeout** *retry-timeout*] [**focalpoint**]

To define a link to an SNA host over an X.25/QLLC connection, perform the following task in global configuration mode:

Task	Command
Define a link to an SNA host over an X.25/QLLC connection.	**sna host** *host-name* **xid-snd** *xid* **x25** *remote-x121-addr* [**qllc** *local-x121-subaddr*] [**interface** *slot/port*] [**window** *window-size*] [**maxiframe** *max-iframe*] [**retries** *retry-count*] [**retry-timeout** *retry-timeout*] [**focalpoint**]

To define a link to an SNA host over a Frame Relay connection, perform the following task in global configuration mode:

Task	Command
Define a link to an SNA host over a Frame Relay connection.	**sna host** *host-name* **xid-snd** *xid* **dlci** *dlci-number* [**rsap** *remote-sap*] [**lsap** *local-sap*] [**interface** *slot/port*] [**window** *window-size*] [**maxiframe** *max-iframe*] [**retries** *retry-count*] [**retry-timeout** *retry-timeout*] [**focalpoint**]

Configure Service Point Support to Use a Data Link Control

To configure Service Point to use a data link control, perform the tasks in one of the following sections:

- Configure Service Point to Use Token Ring, Ethernet, of FDDI
- Configure Service Point to Use RSRB
- Configure Service Point to Use RSRB with Local Acknowledgment
- Configure Service Point to Use Virtual Data-Link Control
- Configure Service Point Support for Frame Relay
- Configure Service Point Support for SDLC
- Configure Service Point Support for X.25

Note You do not need to perform this task if you have configured a DSPU host with the **focalpoint** parameter and have configured the DSPU host to use a data link control.

Configure Service Point to Use Token Ring, Ethernet, of FDDI

To enable a local SAP on the Token Ring, Ethernet, or FDDI interfaces for use by SNA Service Point, perform the following task in interface configuration mode:

Task	Command
Enable local SAP for Service Point.	**sna enable-host** [**lsap** *lsap-address*]

Once a local SAP is enabled, it is ready to accept incoming connection attempts from the remote host. Alternatively, initiate an outgoing connection to the remote host by performing the following task in interface configuration mode:

Task	Command
Initiate connection with host via Token Ring, Ethernet, or FDDI.	**sna start** *host-name*

Configure Service Point to Use RSRB

To define the Service Point/RSRB data link control interface, perform the following tasks in global configuration mode:

Task	Command
Define an RSRB ring group.	**source-bridge ring-group** *ring-group* [*virtual-mac-address*]
Define the Service Point/RSRB interface.	**sna rsrb** *local-virtual-ring bridge-number target-virtual-ring virtual-macaddr*

To enable a local SAP on RSRB for use by hosts, perform the following task in global configuration mode:

Task	Command
Enable local SAP for hosts.	**sna rsrb enable-host** [**lsap** *local-sap*]

Once a local SAP is enabled, it is ready to accept incoming connection attempts from the remote host over RSRB. Alternatively, initiate an outgoing connection to the remote host by performing the following task in global configuration mode:

Task	Command
Initiate connection with host via RSRB.	**sna rsrb start** *host-name*

Configure Service Point to Use RSRB with Local Acknowledgment

To configure Service Point to use RSRB with local acknowledgment, perform the following tasks in global configuration mode:

Task	Command
Define an RSRB ring group.	**source-bridge ring-group** *ring-group* [*virtual-mac-address*]
Define a remote peer with the local acknowledgment feature.	**source-bridge remote-peer** *ring-group* **tcp** *ip-address* **local-ack**
Define the Service Point/RSRB interface.	**sna rsrb** *local-virtual-ring bridge-number target-virtual-ring virtual-macaddr*

Configure Service Point to Use Virtual Data-Link Control

To configure SNA Service Point to use virtual data-link control, you must create an SNA virtual data-link control interface.

Similar to our implementation of SDLLC, the SNA virtual data-link control interface uses the concept of a virtual Token Ring device residing on a virtual Token Ring to represent the Cisco IOS software to upstream hosts and downstream PUs across a network.

Because the upstream host and downstream PU expect their peer to also be on a Token Ring, you must assign a virtual Token Ring address (the SNA virtual data-link control virtual MAC address) to the SNA virtual data-link control interface. Like real Token Ring addresses, the SNA virtual MAC address must be unique across the network.

In addition to assigning the SNA virtual data-link control virtual MAC address, you must also identify the source-route bridging virtual ring number with which the SNA virtual MAC address will be associated. The source-route bridging virtual ring number is set using the **source-bridge ring-group** command, which is documented in the "Source-Route Bridging Commands" chapter of the *Bridging and IBM Networking Command Reference*.

The combination of the SNA virtual MAC address and the source-route bridging virtual ring number identifies the SNA virtual data-link control interface to the rest of the DLSw+ network.

When an end station (either an upstream host or a downstream PU) attempts to connect with the SNA Service Point software, the following events occur:

1. The end station sends explorer packets with the locally administered MAC address on the router interface to which the end station is connected.
2. The router configured with that locally administered MAC address intercepts the frame, DLSw+ adjusts the routing information field (RIF) and sends a response to the end station.
3. The end station establishes a session with the SNA Service Point router.

Prior to creating the SNA virtual data-link control interface, you must also configure DLSw+ peers so that DLSw+ can provide the communication path. The commands for defining DLSw+ local and remote peers are documented in the "DLSw+ Configuration Commands" chapter of the *Bridging and IBM Networking Command Reference*.

To define the Service Point virtual data-link control interface, perform the following task in global configuration mode:

Task	Command
Define the Service Point virtual data-link control interface.	**sna vdlc** *ring-group virtual-mac-address*

After you create the SNA virtual data-link control interface, configure SNA Service Point to use virtual data-link control by enabling a local SAP for upstream connections. To enable a local SAP on virtual data-link control for use by hosts, perform the following task in global configuration mode:

Task	Command
Enable local SAP for hosts.	**sna vdlc enable-host** [**lsap** *local-sap*]

Once a local SAP is enabled, it is ready to accept incoming connection attempts from the remote host using virtual data-link control. Alternatively, initiate an outgoing connection to the remote host by performing the following task in global configuration mode:

Task	Command
Initiate connection with host via virtual data-link control.	**sna vdlc start** *host-name*

Configure Service Point Support for Frame Relay

To configure Service Point support for Frame Relay, perform the following tasks in interface configuration mode:

Task	Command
Define DLCI mapping into LLC2.	**frame-relay map llc2** *dlci-number*
Enable a local SAP for hosts.	**sna enable-host lsap** *lsap-address*

Configure Service Point Support for SDLC

To configure Service Point support for SDLC, perform the following tasks in interface configuration mode:

Task	Command
Specify the SDLC role of the router.	**sdlc role** {**none** | **primary** | **secondary** | **prim-xid-poll**}
Define the SDLC address.	**sdlc address** *hexbyte*
Enable the SDLC address for the host.	**sna enable-host sdlc** *sdlc-address*

Configure Service Point Support for X.25

To configure Service Point support for X.25, perform the following tasks in interface configuration mode:

Task	Command
Define an X.121 address.	**x25 address** *x121-address*
Define remote X.121 address for mapping to QLLC.	**x25 map qllc** *x121-addr*
Enable QLLC subaddress for host.	**sna enable-host qllc** *x121-subaddress*
Enable routing of X.25 packets to serial interface.	**x25 route ^***x121-addr***.* alias** *serial interface*

Specify Names for All Attached LANs

You can specify names for all Token Ring or Ethernet LANs attached to the router. These names are used to identify the LAN when the Cisco IOS software sends an Alert to the host. To specify names for all attached LANs, perform the following tasks in interface configuration mode:

Task	Command
Define the name of an attached LAN.	**lan-name** *lan-name*

Specify the Physical Location of the Router

You can specify the physical location of the router if you intend requesting vital product information from the router. To specify the physical location, perform the following tasks in interface configuration mode:

Task	Command
Define the physical location of the router.	**location** *location-description*

Monitor DSPU and SNA Service Point Feature Status

You can monitor the status of the DSPU and SNA Service Point features. To display information about the state of the DSPU and SNA Service Point features, perform the following tasks in EXEC mode:

Task	Command
Show the status of all DSPU resources.	**show dspu**
Show the status of DSPU hosts or downstream PUs.	**show dspu pu** {*host-name* | *pu-name*} [**all**]
Show the status of a DSPU pool.	**show dspu pool** *pool-name* [**all**]
Show the status of all SNA hosts.	**show sna**
Show the status of an SNA host.	**show sna pu** *host-name* [**all**]

To control the reporting of DSPU notification events (DSPU-specific SNMP Traps and Unsolicited SNA Messages to Operator), perform the following command in global configuration mode:

Task	Command
Specify the level of notification event reporting.	**dspu notification-level** {**off** | **low** | **medium** | **high**}

DSPU and SNA Service Point Configuration Examples

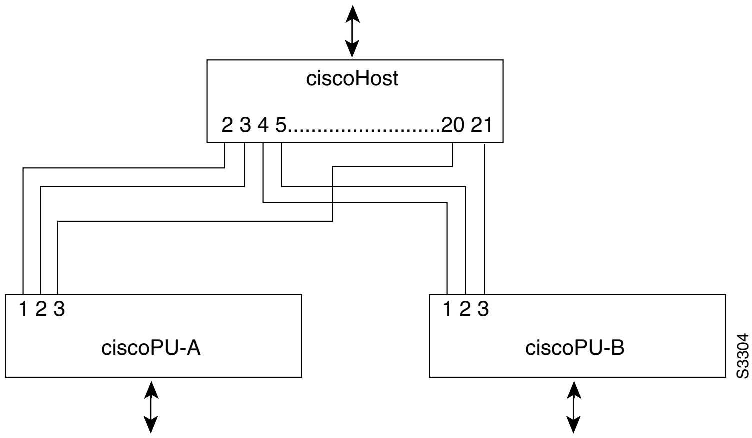
The following sections provide DSPU and SNA Service Point configuration examples:

- Dedicated LU Routing Example
- Pooled LU Routing Example
- Upstream Host via RSRB DSPU Configuration Example
- DSPU over DLSw+ using Virtual Data-Link Control Configuration Example
- Downstream PU via SDLC DSPU Configuration Example
- Upstream Host via SDLC DSPU Configuration Example
- Downstream PU via QLLC/X.25 DSPU Configuration Example
- Upstream Host via Frame Relay DSPU Configuration Example
- DSPU NCIA Configuration Example
- SNA Service Point Support Configuration Example
- SNA Service Point over DLSw+ using Virtual Data-Link Control Configuration Example

Dedicated LU Routing Example

Figure 115 illustrates the use of dedicated LU routing. Each upstream host LU is dedicated for use by a specific downstream LU.

Figure 115 Dedicated LU Routing



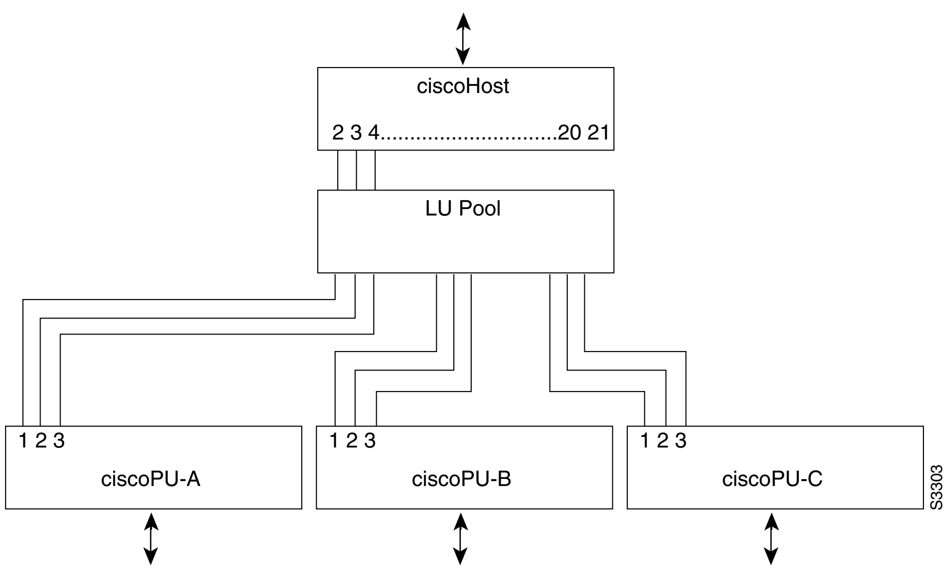
The following is a configuration file for the dedicated LU routing shown in Figure 115:

```
dspu host ciscohost xid-snd 06500001 rmac 4000.3745.0001
dspu pu ciscopu-a xid-rcv 05D00001 rmac 1000.5AED.0001
dspu lu 1 2 host ciscohost 2
dspu lu 3 3 host ciscohost 20
dspu pu ciscopu-b xid-rcv 05D00002 rmac 1000.5AED.0002
dspu lu 1 2 host ciscohost 4
dspu lu 3 3 host ciscohost 21
```

Pooled LU Routing Example

Figure 116 illustrates the use of pooled LU routing. Each upstream LU is configured in the LU pool and each downstream LU is configured as a pooled LU.

Figure 116 Pooled LU Routing



The following is a configuration file for the pooled LU routing shown in Figure 116:

```
dspu host ciscohost xid-snd 06500001 rmac 4000.3745.0001
dspu pool lupool host ciscohost lu 2 21
dspu pu ciscopu-a xid-rcv 05D00001 rmac 1000.5AED.0001
dspu lu 1 3 pool lupool
dspu pu ciscopu-b xid-rcv 05D00002 rmac 1000.5AED.0002
dspu lu 1 3 pool lupool
dspu pu ciscopu-c xid-rcv 05D00003 rmac 1000.5AED.0003
dspu lu 1 3 pool lupool
```

Upstream Host via RSRB DSPU Configuration Example

The following configuration example represents one possible definition for the network topology shown earlier in Figure 115. This example demonstrates the configuration of an upstream host via RSRB (with local acknowledgment) and downstream PUs via Token Ring.

```
source-bridge ring-group 99
source-bridge remote-peer 99 tcp 150.10.13.1
source-bridge remote-peer 99 tcp 150.10.13.2 local-ack

dspu rsrb 88 1 99 4000.ffff.0001
dspu rsrb enable-host lsap 4

dspu host ciscohost xid-snd 06500001 rmac 4000.3172.0001 rsap 4 lsap 4
dspu pool ciscopool host ciscohost lu 2 8
dspu rsrb start ciscohost

dspu pu ciscopu1 xid-rcv 05d00001
dspu lu 2 3 pool ciscopool

dspu pu ciscopu2 xid-rcv 05d00002
dspu lu 2 4 pool ciscopool

dspu pu ciscopu3 xid-rcv 05d00003
dspu lu 2 2 pool ciscopool

dspu pu ciscopu4 xid-rcv 05d00004
dspu lu 2 2 pool ciscopool
dspu lu 3 3 host ciscohost 9

interface tokenring 0
 description tokenring connection for downstream PUs
 ring-speed 16
 dspu enable-pu lsap 8
```

DSPU over DLSw+ using Virtual Data-Link Control Configuration Example

This configuration example illustrates pooled LU routing over DLSw+ using virtual data-link control.

```
source-bridge ring-group 99
dlsw local-peer peer-id 150.10.16.2
dlsw remote-peer 0 tcp 150.10.16.1
!
dspu vdlc 99 4000.4500.01f0
dspu vdlc enable-pu lsap 8
dspu vdlc enable-host lsap 12
!
dspu host HOST-B xid-snd 065bbbb0 rmac 4000.7000.01f1 rsap 4 lsap 12 focalpoint
dspu pool pool-b host HOST-B lu 2 254
!
```

```
dspu pu PU3K-A xid-rcv 05d0000a rmac 4000.3000.0100 rsap 10 lsap 8
dspu lu 2 254 pool pool-b
!
dspu default-pu
dspu lu 2 5 pool pool3k-a
!
dspu vdlc start HOST-B
dspu vdlc start PU3K-A
!
interface serial 3
 description IP connection to dspu7k
 ip address 150.10.16.2 255.255.255.0
 clockrate 4000000
```

Downstream PU via SDLC DSPU Configuration Example

This example demonstrates the configuration of downstream PUs via SDLC and an upstream host via Token Ring.

```
dspu host ciscohost xid-snd 06500001 rmac 4000.3172.0001 rsap 4 lsap 12
dspu pool ciscopool host ciscohost lu 2 11
!
dspu pu pu-sdlc0 sdlc C1 interface serial 0
dspu lu 2 6 pool ciscopool
!
dspu pu pu-sdlc1 sdlc C1 interface serial 1
dspu lu 2 6 pool ciscopool
!
interface serial 0
 description SDLC connection for pu-sdlc0
 encapsulation sdlc
 sdlc role primary
 sdlc address C1
 dspu enable-pu sdlc C1
 clockrate 56000
!
interface serial 1
 description SDLC connection for pu-sdlc1
 encapsulation sdlc
 sdlc role primary
 sdlc address C1
 dspu enable-pu sdlc C1
 clockrate 56000
!
interface tokenring 0
 description tokenring connection for ciscohost
 ring-speed 16
 dspu enable-host lsap 12
 dspu start ciscohost
```

Upstream Host via SDLC DSPU Configuration Example

This example demonstrates the configuration of an upstream host via SDLC and downstream PUs via Token Ring and Ethernet.

```
dspu host ciscohost xid-snd 06500001 sdlc C1 interface serial 0
dspu pool ciscopool host ciscohost lu 2 11
!
dspu pu pu-token rmac 4000.4444.0001 rsap 4 lsap 8
dspu lu 2 6 pool ciscopool
!
```

```
dspu pu pu-ether rmac 0200.2222.0001 rsap 4 lsap 8
dspu lu 2 6 pool ciscopool
!
interface serial 0
 description SDLC connection for ciscohost
 encapsulation sdlc
 sdlc role secondary
 sdlc address C1
 dspu enable-host sdlc C1
 clockrate 56000
!
interface tokenring 0
 description tokenring connection for pu-token
 ring-speed 16
 dspu enable-pu lsap 8
!
interface ethernet 0
 description Ethernet connection for pu-ether
 dspu enable-pu lsap 8
```

Downstream PU via QLLC/X.25 DSPU Configuration Example

This example demonstrates the configuration of a downstream PU via QLLC/X.25 and upstream host via Ethernet.

```
x25 routing
!
dspu host ciscohost xid-snd 06500001 rmac 0200.2222.0001 rsap 4 lsap 12
dspu pool ciscopool host ciscohost lu 2 11
!
dspu pu pu-qllc x25 320108 qllc 08
dspu lu 2 11 pool ciscopool
!
interface serial 0
 description QLLC connection for pu-qllc
 encapsulation x25
 x25 address 3202
 x25 map qllc 320108
 dspu enable-pu qllc 8
!
interface ethernet 0
 description Ethernet connection for pu-ether
 dspu enable-host lsap 12
 dspu start ciscohost
!
 x25 route ^3202.* alias serial 0
```

Upstream Host via Frame Relay DSPU Configuration Example

This example demonstrates the configuration of an upstream host via Frame Relay and downstream PUs via Token Ring and Ethernet.

```
dspu host ciscohost xid-snd 06500001 dlci 200 rsap 4 lsap 12
dspu pool ciscopool host ciscohost lu 2 11
!
dspu pu pu-token rmac 4000.4444.0001 rsap 4 lsap 8
dspu lu 2 6 pool ciscopool
!
dspu pu pu-ether rmac 0200.2222.0001 rsap 4 lsap 8
dspu lu 2 6 pool ciscopool
!
```

```
interface serial 0
 description Frame Relay connection for ciscohost
 encapsulation frame-relay ietf
 frame-relay map llc2 200
 dspu enable-host lsap 12
 dspu start ciscohost
!
interface tokenring 0
 description tokenring connection for pu-token
 ring-speed 16
 dspu enable-pu lsap 8
!
interface ethernet 0
 description Ethernet connection for pu-ether
 dspu enable-pu lsap 8
```

DSPU NCIA Configuration Example

This configuration example illustrates an NCIA client/server session using DSPU.

```
ncia server 1 10.2.20.4 4000.3745.0001 1000.0000.0001 128
!
dspu ncia 1
dspu ncia enable-pu lsap 8
!
dspu host HOST-9370 xid-snd 11100001 rmac 4000.1060.1000 rsap 4 lsap 4
!
dspu pu CISCOPU-A xid-rcv 01700001
dspu lu 2 6 host HOST-9370 2
!
interface TokenRing 0
 ring-speed 16
 llc2 xid-retry-time 0
 dspu enable-host lsap 4
 dspu start HOST-9370
!
```

SNA Service Point Support Configuration Example

The following is an example of an RSRB configuration that implements SNA Service Point:

```
source-bridge ring-group 99
source-bridge remote-peer 99 tcp 150.10.13.2 local-ack
!
sna rsrb 88 1 99 4000.ffff.0001
!
sna host CNM02 xid-snd 05dbc000 rmac 4001.3745.1088 rsap 4 lsap 4 focalpoint
sna rsrb enable-host lsap 4
sna rsrb start CNM02
!
```

SNA Service Point over DLSw+ using Virtual Data-Link Control Configuration Example

The following is an example of an SNA Service Point configuration that uses virtual data-link control over DLSw+:

```
source-bridge ring-group 99
dlsw local-peer peer-id 150.10.16.2
dlsw remote-peer 0 tcp 150.10.16.1
!
```

```
sna vdlc 99 4000.4500.01f0
sna vdlc enable-host lsap 12
!
sna host HOST-B xid-snd 065bbbb0 rmac 4000.7000.01f1 rsap 4 lsap 12 focalpoint
!
sna vdlc start HOST-B
!
interface serial 3
 description IP connection to dspu7k
 ip address 150.10.16.2 255.255.255.0
 clockrate 4000000
```