CISCO SYSTEMS

Cisco VCO/4K SS7 Subsystem, ITU V5.2, Manual

December 2000

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

Part Number: 78-xxxxx-01

Cisco VCO/4K SS7 Subsystem, ITU V5.2, Manual

SECTION 1 – SS7 SUBSYSTEM OVERVIEW

SECTION 2 – INSTALLATION

SECTION 3 – SOFTWARE INSTALLATION AND NETWORK CONFIGURATION

SECTION 4 – SS7 SUBSYSTEM CONFIGURATION

SECTION 5 – SYSTEM ADMINISTRATION

SECTION 6 – ISUP

SECTION 7 – TCAP

Preface

The *Cisco VCO/4K SS7 Subsystem, ITU V5.2, Manual* provides detailed information about how to use the SS7 subsystem hardware and software to integrate Signaling System No 7 (SS7) networks with VCO/4K Systems. This manual is organized as follows:

- **Section 1**—Contains a product overview
- **Section 2**—Contains instructions for installing the system hardware
- **Section 3**—Describes how to unpack the software with the sys-config script and how to configure the system on the network
- **Section 4**—Describes how to configure the SS7 subsystem components
- **Section 5**—Describes how to manage the SS7 subsystem
- **Section 6**—Describes ISUP applications
- **Section 7**—Describes TCAP applications
- **Appendix A**—Lists basic UNIX/vi editor basics
- **Appendix B**—Describes the file structure of the SS7 subsystem software
- **Appendix C**—Contains instructions for upgrading/re-installing the SS7 subsystem operating platform
- **Appendix D**—Contains instructions for upgrading/re-installing the SS7 subsystem software
- **Appendix E**—Lists the ISUP message types and parameters
- **Appendix F**—Lists country variants

For information about the enhancements, problems corrected, special considerations, and known functional constraints, refer to the *Cisco VCO/4K SS7 ISUP, ITU V5.2, Release Notes*.

This manual is intended for programmers familiar with the VCO/4K Systems, SS7 concepts, UNIX, and Ethernet.

For more information on basic SS7 concepts, refer to *SS7 Fundamentals*.

Obtaining Documentation

World Wide Web

You can access the most current Cisco documentation on the World Wide Web at http://www.cisco.com, http://www-china.cisco.com, or http://www-europe.cisco.com.

Documentation CD-ROM

Cisco documentation and additional literature are available in a CD-ROM package, which ships with your product. The Documentation CD-ROM is updated monthly. Therefore, it is probably more current than printed documentation. The CD-ROM package is available as a single unit or as an annual subscription.

Ordering Documentation

Registered CCO users can order the Documentation CD-ROM and other Cisco Product documentation through our online Subscription Services at http://www.cisco.com/cgi-bin/subcat/kaojump.cgi.

Nonregistered CCO users can order documentation through a local account representative by calling Cisco's corporate headquarters (California, USA) at 408 526-4000 or, in North America, call 800 553-NETS (6387).

Obtaining Technical Assistance

Cisco provides Cisco Connection Online (CCO) as a starting point for all technical assistance. Warranty or maintenance contract customers can use the Technical Assistance Center. All customers can submit technical feedback on Cisco documentation using the web, e-mail, a self-addressed stamped response card included in many printed docs, or by sending mail to Cisco.

Cisco Connection Online

Cisco continues to revolutionize how business is done on the Internet. Cisco Connection Online is the foundation of a suite of interactive, networked services that provides immediate, open access to Cisco information and resources at anytime, from anywhere in the world. This highly integrated Internet application is a powerful, easy-to-use tool for doing business with Cisco.

CCO's broad range of features and services helps customers and partners to streamline business processes and improve productivity. Through CCO, you will find information about Cisco and our networking solutions, services, and programs. In addition, you can resolve technical issues with online support services, download and test software packages, and order Cisco learning materials and merchandise. Valuable online skill assessment, training, and certification programs are also available.

Customers and partners can self-register on CCO to obtain additional personalized information and services. Registered users may order products, check on the status of an order and view benefits specific to their relationships with Cisco.

You can access CCO in the following ways:

- WWW: www.cisco.com
- Telnet: cco.cisco.com
- Modem using standard connection rates and the following terminal settings: VT100 emulation; 8 data bits; no parity; and 1 stop bit.
  - From North America, call 408 526-8070
  - From Europe, call 33 1 64 46 40 82

You can e-mail questions about using CCO to cco-team@cisco.com.

Technical Assistance Center

The Cisco Technical Assistance Center (TAC) is available to warranty or maintenance contract customers who need technical assistance with a Cisco product that is under warranty or covered by a maintenance contract.

To display the TAC web site that includes links to technical support information and software upgrades and for requesting TAC support, use www.cisco.com/techsupport.

To contact by e-mail, use one of the following:

Language	E-mail Address
English	tac@cisco.com
Hanzi (Chinese)	chinese-tac@cisco.com
Kanji (Japanese)	japan-tac@cisco.com
Hangul (Korean)	korea-tac@cisco.com
Spanish	tac@cisco.com
Thai	thai-tac@cisco.com

In North America, TAC can be reached at 800 553-2447 or 408 526-7209. For other telephone numbers and TAC e-mail addresses worldwide, consult the following web site: http://www.cisco.com/warp/public/687/Directory/DirTAC.shtml.

Documentation Feedback

If you are reading Cisco product documentation on the World Wide Web, you can submit technical comments electronically. Click **Feedback** in the toolbar and select **Documentation**. After you complete the form, click **Submit** to send it to Cisco.

You can e-mail your comments to bug-doc@cisco.com.

To submit your comments by mail, for your convenience many documents contain a response card behind the front cover. Otherwise, you can mail your comments to the following address:

Cisco Systems, Inc.
Document Resource Connection
170 West Tasman Drive
San Jose, CA 95134-9883

We appreciate and value your comments.

Section 1
SS7 SUBSYSTEM OVERVIEW

1.1 INTRODUCTION

The SS7 subsystem serves as a gateway between Signaling System No. 7 (SS7) intelligent networks and Public Switched Telephone Networks (PSTN) by integrating the SS7 network with a VCO/4K system and host computer(s). This integration can significantly improve call processing because the call setup information arrives independent of the voice traffic. Independent signaling increases the call processing rate by a minimum of 60% over the rates of other signaling methods. SS7 integration also reduces network congestion and deployment costs through greater port utilization and alternative routing capabilities.

The SS7 subsystem allows the service provider to implement the following enhanced services:

- ISDN User Part (ISUP) services for call processing and switching, such as:
  - *Operator Services System support*
  - *ISDN or Feature Group D Interworking*
  - *Personal Communication Services (PCS)/Enhanced Services Billing*
  - *Integration of full Service Switching Point (SSP) functionality on the VCO/4K*
  - *Wireless and wire line internetworking*
- Transactions Capabilities Applications Part (TCAP) applications, such as:
  - *Credit card/debit card validation*
  - *Personal number/"follow me" services*
  - *Network-based services including 800 number routing and Automatic Call Distribution (ACD)*
  - *Mediated access*
  - *Cellular roaming services*

The SS7 subsystem consists of both hardware and software components that are installed in a VME SS7 shelf that can be mounted in a 19-inch utility rack with a VCO/4K system. An SS7 subsystem can support one VCO/4K system and up to eight host computers that all reside on Ethernet LAN.

Figure 1.1 is an illustration of an SS7 subsystem configured with a VCO/4K and four host computers. The SS7 subsystem connects directly to the Ethernet LAN and SS7 network.

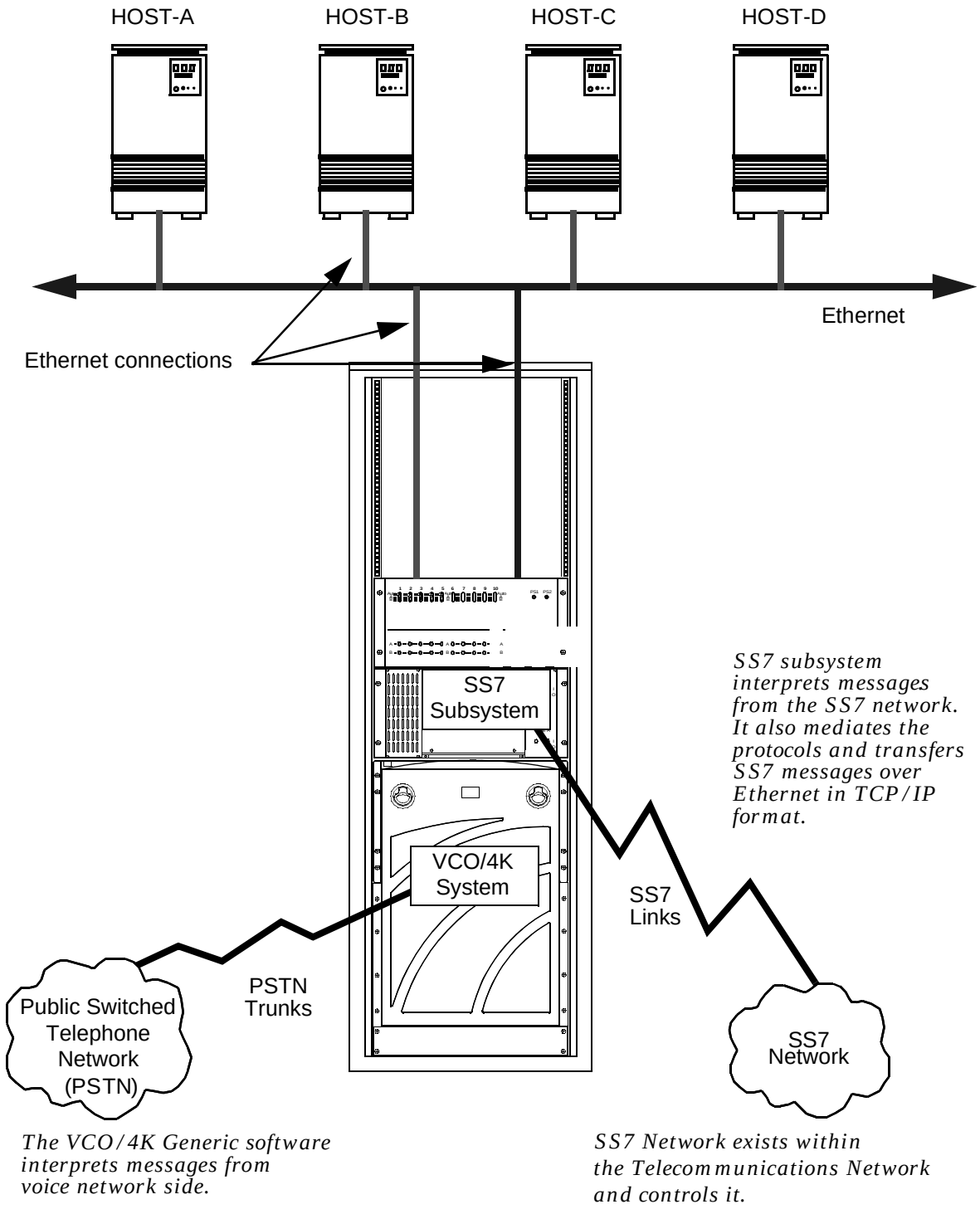


Figure 1.1: Sample SS7 Subsystem Configuration

1.2 SS7 SUBSYSTEM HARDWARE

A non-redundant SS7 subsystem consists of the following hardware components:

- One SPARC CPU/Sbus Assembly
- One Storage Subsystem (3.5-inch 1.4MB floppy disk and 1GB, 2GB, or 4GB hard drive with SCSI interfaces)
- One VME SS7 Shelf
- One external CD-ROM drive
- One modem
- One administrative console

NOTE: The modem and administrative console can be customer supplied.

The CPU/Sbus Assembly runs at 110 MHz and is equipped with 32MB or 64MB RAM. Ports on the card's front panel support connections to the SS7 subsystem peripherals and SS7 network.

The hardware includes an external CD-ROM drive so that the SS7 subsystem software can be reinstalled if the factory-installed version is damaged or corrupted. The power supply in the CD-ROM drive is a universal power supply with 100–120/200–220 VAC, and 47–63 hertz. The external CD-ROM drive does not need to be connected to the system for normal operation.

For redundant systems, an SS7 subsystem consists of the following hardware components:

- Two SPARC CPU/Sbus Assemblies
- Two Storage Subsystems (3.5-inch 1.4MB floppy disk and 1GB, 2GB, or 4GB hard drive with SCSI interfaces)
- One VME SS7 Shelf
- One external CD-ROM drive
- One SS7 selector switch for redundancy
- One modem/A/B transfer switch
- Two administrative consoles (or one dual session console)

NOTE: The modem/A/B transfer switch and administrative console can be customer supplied.

Modem support and console operation can also be done using Telnet sessions.

Note that the SS7 subsystem hardware for redundant configurations includes an SS7 selector switch, which provides SS7 link redundancy for both sides of the redundant SS7 subsystem. The selector switch is a -48 VDC device, designed to mount in a 19-inch utility rack separate from the VCO/4K. When installing in a site with AC power only, an AC to DC converter is required. (The converter is supplied by the customer.)

CAUTION: Do not mount the SS7 selector switch in the VCO/4K or power it with the VCO/4K's power supply.

The hardware for a redundant SS7 subsystem configuration includes one external CD-ROM drive and one modem. The external CD-ROM drive can be connected to either side of the configuration without affecting the operation of the system. The modem is connected to both sides of the redundant system by an electronic A/B transfer switch (which you must purchase separately).

A single ASCII terminal can also be connected to both sides with a second electronic A/B transfer switch. However, the transfer switch must be capable of providing surgeless, spikeless change-overs. If the transfer switch does not have this capability, the SS7 subsystem may abort and return to the boot monitor when change overs occur.

1.3 SS7 SUBSYSTEM SOFTWARE

From a software point of view, the SS7 subsystem sits between the host computer and the VCO/4K system on the Ethernet as shown in Figure 1.2. To the host application, the SS7 subsystem is the server. However, to the VCO/4K generic software, the SS7 subsystem is a client. The SS7 subsystem also provides the interface to the SS7 Network.

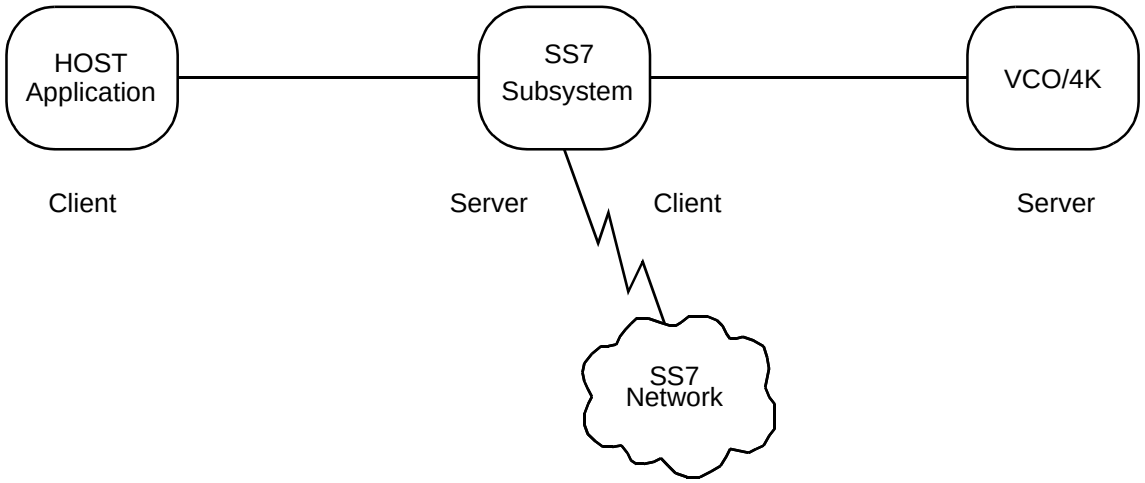


Figure 1.2: SS7 Subsystem from a Software Point of View

The SS7 subsystem software supports up to eight TCP connections as shown in Figure 1.3. These are logical links, not physical links.

CAUTION: Additional TCP connections affect VCO/4K call handling capacity. Optimal performance can be achieved with four or fewer simultaneously active TCP connections.

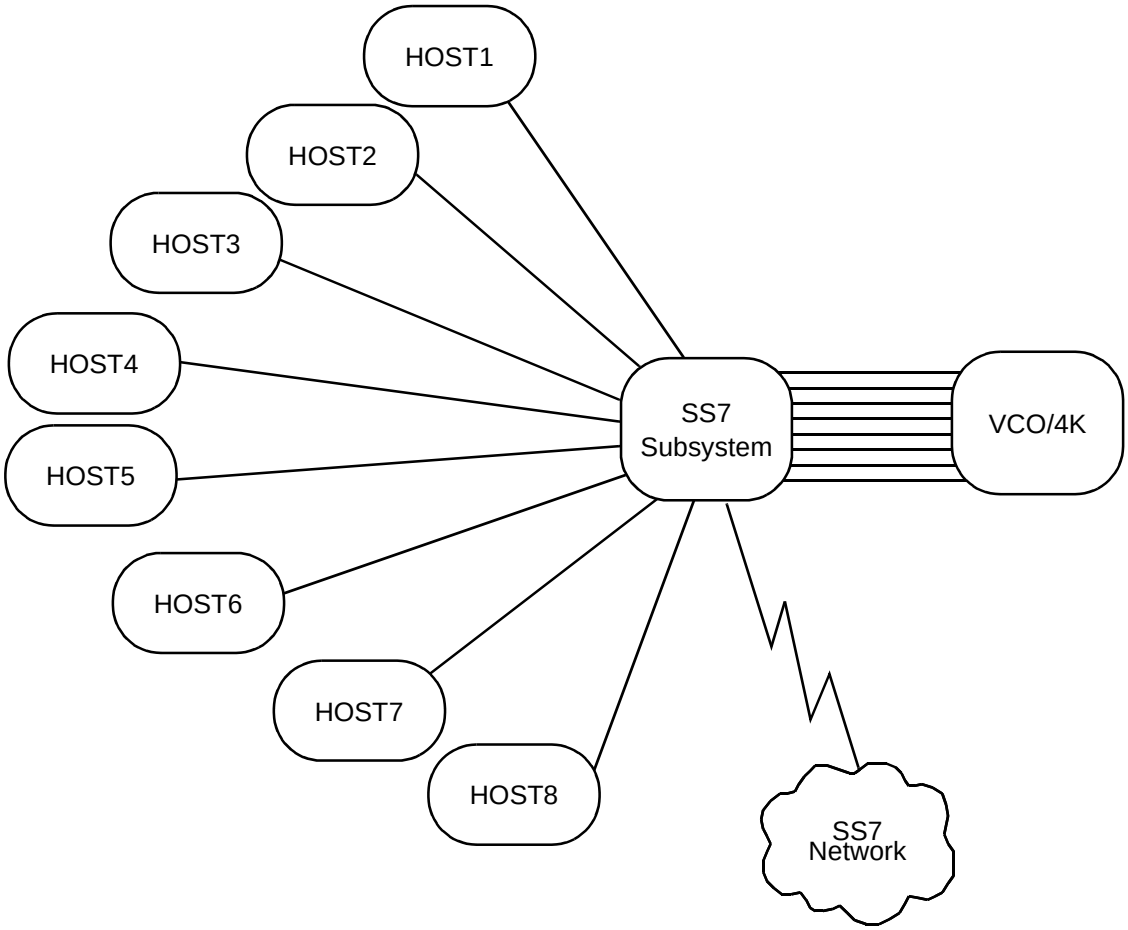


Figure 1.3: Software Configuration Example with Eight Hosts

For each logical link between the host and the SS7 subsystem, there also is a logical link between the SS7 subsystem and the VCO/4K. This provides an end-to-end communication path between each host and the VCO/4K.

At initialization, the SS7 subsystem waits for a connection request from a host. Once a host connection is established, the SS7 subsystem sends a connection request to the VCO/4K.

The hosts share the incoming SS7 call load in either a ROUNDROBIN or BROADCAST mode. In ROUNDROBIN mode, the next host computer in line becomes the controlling host. In BROADCAST mode, the SS7 subsystem sends a report to all active host computers and a host takes control of the call port depending on the call port addresses specified in the host's application program.

SS7 messages from the host are sent to SS7 subsystem software through commands with a destination VCA of $C0. SS7 messages from the network are translated into host reports with a source VCA of $C0.

1.3.1 SS7 Subsystem Software Components

The SS7 subsystem software is installed on the SS7 subsystem's hard drive and consists of the following components:

- Solaris OS
- Software modules that perform the actual interworking processes
- Software modules that interface between the SS7 network, VCO/4K, and the host computer
- Software modules that provide configuration, control, and administration functions

Figure 1.4 is an illustration of the SS7 subsystem software architecture.

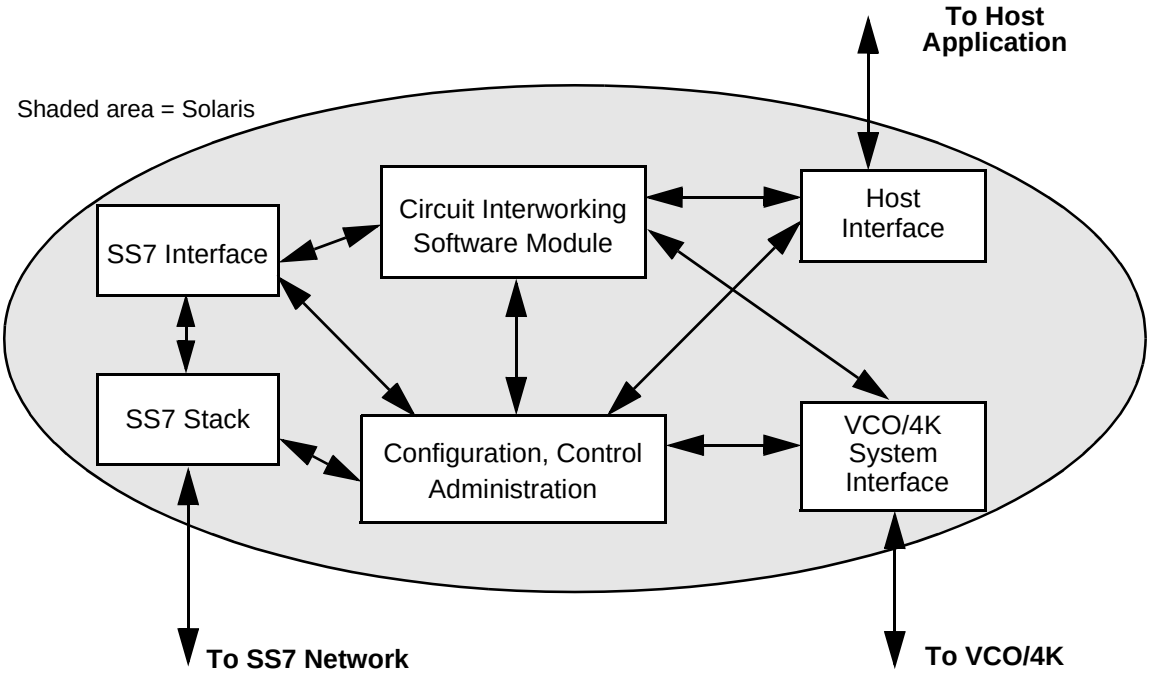


Figure 1.4: SS7 Subsystem Software Components

SS7 Stack

Provides the data link for signaling. The stack consists of three layers that map one for one to the first three layers of the standard SS7 protocol (Message Transfer Part (MTP) Level 1, MTP Level 2, and MTP Level 3). For further information on the function of these stacks, refer to *Section 4.4*.

Circuit Interworking Software Module

Translates SS7 signals for the VCO/4K and host applications and performs call processing and circuit maintenance tasks. SS7 messages are encapsulated and presented to the host system in VCO/4K or SS7 commands and reports.

During call processing, the Circuit Interworking software acts upon the call setup and call clearing messages. The rest of the call processing is the responsibility of the host application. The Circuit Interworking software transmits information to the host, it translates all SS7 messages into SS7 Network Message Reception ($EA) reports. The host application is then responsible for generating SS7 messages using the SS7 Network Message Generation($49) command.

The host application, through the SS7 $49 Command, is also responsible for generating the SS7 call acceptance message (ACM), SS7 answer message (ANM), and the SS7 call clearing message (REL).

Circuit Interworking software supports SS7-to-SS7 calls, SS7-to-VCO/4K calls, and VCO/4K-to-SS7 calls.

Interface Software Modules (SS7 Interface—Host Interface—VCO/4K System Interface)

The SS7 subsystem software includes interface software modules for the SS7 network, VCO/4K, and host computer. All modules operate over Ethernet using TCP/IP.

When the VCO/4K Interface software module receives a report from the VCO/4K, the report is passed to the Circuit Interworking software. Call control commands are also transmitted to the VCO/4K by the Circuit Interworking software through the VCO/4K's Interface.

The SS7 Network Interface software module manages the interface to SS7 links on the SS7 subsystem. SS7 messages from the network are transmitted to the Circuit Interworking software through this interface. SS7 messages generated by the Circuit Interworking software to the SS7 network also pass through the SS7 Network Interface software module.

The Host Interface software module manages the interface to the host computer. This interface supports the transmission of both SS7 and VCO/4K-related event reports to the host, as well as receipt of command messages destined for either the SS7 network or the VCO/4K. All received messages are passed to the Circuit Interworking software module, where they are processed by the software module or passed to the SS7 network or VCO/4K.

Configuration, Control, and Administration

The SS7 subsystem configuration data contains the information necessary for the Circuit Interworking software module to interface with the VCO/4K, host computer, and SS7 network.

The SS7 subsystem resource provisioning data contains the circuit IDs and the circuit groups.

The SS7 subsystem call processing configuration data dictates the content of SS7 messages that are generated and transmitted to the SS7 network by the Circuit Interworking software.

Section 2
INSTALLATION

2.1 INTRODUCTION

This section describes how to install the SS7 subsystem hardware in a VCO/4K system. This section is organized as follows:

- *Section 2.2* is a list of the tasks that must be completed for an SS7 subsystem hardware installation.
- *Section 2.3* is a list of hardware components. Before you begin the installation, verify that you have received all components. If components are missing, contact Cisco Systems Technical Support.
- *Section 2.4* describes how to install the VME SS7 Shelf. The SS7 Shelf is an external VME enclosure for the SS7 subsystem hardware (SPARC CPU/Sbus Assembly and SS7 Storage Subsystem). It allows VCO/4K customers to implement SS7 applications without reducing the number of slots available for port interface and service circuit cards.
- *Section 2.5* describes how to install the cards in non-redundant systems or Side A of redundant systems. This subsection also describes how to physically connect the links to the SS7 network.
- *Section 2.6* describes how install the ASCII terminal, modem, and CD-ROM on non-redundant systems. This completes the installation on non-redundant systems. *If you have a redundant system, skip this subsection.*
- *Section 2.7* describes how to install the cards in Side B of a redundant system, and how to connect the Side B links to the SS7 network.
- *Section 2.8* describes how to install the ASCII terminals, modem, and CD-ROM on redundant systems.
- *Section 2.9* describes how to install the SS7 selector switch for redundant systems. The SS7 selector switch provides SS7 link redundancy connection to sides A and B of the SS7 subsystem.

2.2 INSTALLATION CHECKLIST

This subsection summarizes the tasks that must be completed for an SS7 subsystem hardware installation. Items are listed in the order they should be completed.

> *NOTE: This list is independent of installing the VCO/4K system. The VCO/4K system should be completely installed prior to an SS7 installation.*

Hardware Installation Task List

1. Verify hardware components (see *Section 2.3*).
2. Install hardware into the VME shelf:
    - CPU
    - Hard drive
    - Back plane ribbon
    - Verify jumper settings
    - Mount VME SS7 shelf and fall back switch
    - Power VME SS7 shelf and fall back switch
3. Install cables:
    - Data links
    - Fall back switch
    - Ethernet/LAN
    - Console/status control and Y-cable
    - Modem (if used)

    NOTE: Cable installation instructions in this manual apply to non-SVX environments only.
4. Verify configuration information (refer to SS7 Facilities Planning Checklist):
    - IP address(es)*
    - SPARC Hostname(s)*
    - VCO/4K hostname(s*)
    - Host socket(s)*
    - Name service, domain, subnet (check with LAN administrator)
    - SS7 point codes—Cisco, STPs, Tandem switch
    - CICs
    - Route set names
    - Link set names
    - Span locations on VCO/4K
5. Connect console to A-side SPARC and boot A-side SPARC.

    ** Need two for a redundant system, or one for a non-redundant system.*

2.3 VERIFYING HARDWARE COMPONENTS

Your hardware depends on whether you ordered a non-redundant or redundant system.

2.3.1 Non-Redundant Systems

If you ordered a non-redundant system, you should have received the following components:

- SPARC CPU5V/Sbus assembly
- Storage Subsystem
- One P2 ribbon cable
- One cable with micro D-26 to DB-25 connectors
- One two-way splitter cable with DB-25 to DB-25 connectors
- Either four (4-link) or eight (8-link) SS7 network link cables with micro D-26 to DB-37 connectors
- One three-way splitter cable with DB-37 to three DB-25s
- External CD-ROM drive with SCSI cable

  NOTE: The power supply in the CD-ROM drive is a universal power supply with 100–120/200–220 VAC, and 47–63 hertz.

- One VME SS7 Shelf and a NEBS bracket mounting kit

  NOTE: The VME shelf has an internal cable harness for connection to ports on the CPU5V to the rear connection panel.

In addition to these components, a non-redundant system requires the following components, which you must purchase separately:

- Ethernet AUI cable or LAN transceiver

  *NOTE: You **must** use shielded cables for your Ethernet AUI connections.*

- Modem (for remote maintenance)/one 25-foot EIA/TIA-232 cable with DB-25 to DB-25 connectors

  *NOTE: **There is no modem capability with an 8-link system**. Instead, you must arrange dial-up to a local network and Telnet access to the SS7 subsystem.*

- ASCII terminal with cable
- Null modem (essential for installation/configuration)

  CAUTION: Attaching a terminal to the SS7 SPARC serial port can cause an SS7 reboot/halt under certain failure conditions. This complication is in the Solaris operating system, which hangs when the VT console is either powered down or has a power supply failure.

  Cisco Systems recommends disconnecting the VT terminal(s) from the SPARC processor serial connection and either (a) reconnecting for active use only, or (b) leaving disconnected and performing all terminal functions via a Telnet session from the host or another connectivity device.

- 19-inch utility rack
- –48 VDC power source

  NOTE: If your site has AC power only, an AC-to-DC rectifier of sufficient output power must be used.

2.3.2 Redundant Systems

If you ordered a redundant system, you should have received the following components:

- Two SPARC CPU5V/Sbus assemblies
- Two Storage Subsystems
- Two P2 ribbon cables
- Two cables with micro D-26 to DB-25 connectors
- Two two-way splitter cables with DB-25 to DB-25 connectors
- Either eight (4-link) or 16 (8-link) SS7 network link cables with micro D-26 to DB-37 connectors
- Two three-way splitter cables with DB-37 to DB-25 connectors
- External CD-ROM Drive with SCSI cable

  NOTE: The power supply in the CD-ROM drive is a universal power supply with 100–120/200–220 VAC, and 47–63 hertz.

- One VME SS7 Shelf and a NEBS bracket mounting kit

  NOTE: The VME shelf has an internal cable harness for connection to ports on the CPU5V to the rear connection panel.

- One SS7 Selector Switch
- Eight cables with DB-25 to DB-37 connectors
- Four cables with DB-25 to DB-37 connectors
- One EBS Status/Control cable assembly

In addition to these components, a redundant installation requires the following components, which you must purchase separately:

- Two Ethernet AUI cable or LAN transceivers

  *NOTE: You **must** use shielded cables for your Ethernet AUI connections.*

- Modem (for remote maintenance)/two 25-foot EIA/TIA-232 cables with DB-25 to DB-25 connectors

  NOTE: Cisco Systems recommends a separate modem to each side. Otherwise, the modem connection can hang periodically and be unusable.

- A/B transfer switch for single modem (available from Cisco Systems)

  *NOTE: **There is no modem capability with an 8-link system**. Instead, you must arrange dial-up to a local network and Telnet access to the SS7 subsystem.*

- Two ASCII terminals with cables
- Two null modems (essential for installation/configuration), or one null modem and one ASCII terminal with dual session capability (available from Cisco Systems)

  CAUTION: Attaching a terminal to the SS7 SPARC serial port can cause an SS7 reboot/halt under certain failure conditions. This complication is in the Solaris operating system, which hangs when the VT console is either powered down or has a power supply failure.

  Cisco Systems recommends disconnecting the VT terminal(s) from the SPARC processor serial connection and either (a) reconnecting for active use only, or (b) leaving disconnected and performing all terminal functions via a Telnet session from the host or another connectivity device.
- 19-inch utility rack for the SS7 Selector Switch and VME SS7 Shelf
- Two –48 VDC power sources for the SS7 Selector Switch and VME SS7 Shelf

  NOTE: If your site has AC power only, an AC-to-DC rectifier of sufficient output power must be used.

2.4 INSTALLING THE SS7 SHELF: VCO/4K SYSTEMS

This subsection describes how to install the VME SS7 Shelf.

2.4.1 Specifications

Part Number:	Contact your Cisco Systems sales representative
Dimensions:	Height — 8.75 inches (22.23 cm) Depth — 20 inches (50.8 cm) Width —19 inches (48.26 cm) Designed to mount in a 19” rack
Power Input:	–48 VDC — dual input
Current:	5.0 amps maximum, 2.5 amps typical per input
Input Fuse Size:	10 amps/250 volts
Compliance:	EMI/EMC FCC Part 15 (U.S. and Canada) EN55022/50082 (for Europe) NEBS GR-63-CORE (Issue 1, 1995) GR-1089-CORE (Issue 2, 1997) UL 1950 CSA C22.2 EN60950 (for Europe) IEC-950
Weight (max.):	40 lbs.
Operational Temperature Range:	40 to 100°F 10 to 40°C
Operational Temperature Gradient:	15°F (10°C) per hour
Operational Humidity:	20 to 80 percent (%), non-condensing
Operational Altitude:	0 to 10,000 ft 0 to 3,048 m

2.4.2 Installation

To install the VME SS7 shelf, complete the following steps:

1. Locate the VME SS7 shelf. Open the rear door by unscrewing the four captive fastener screws (see Figure 2.1).

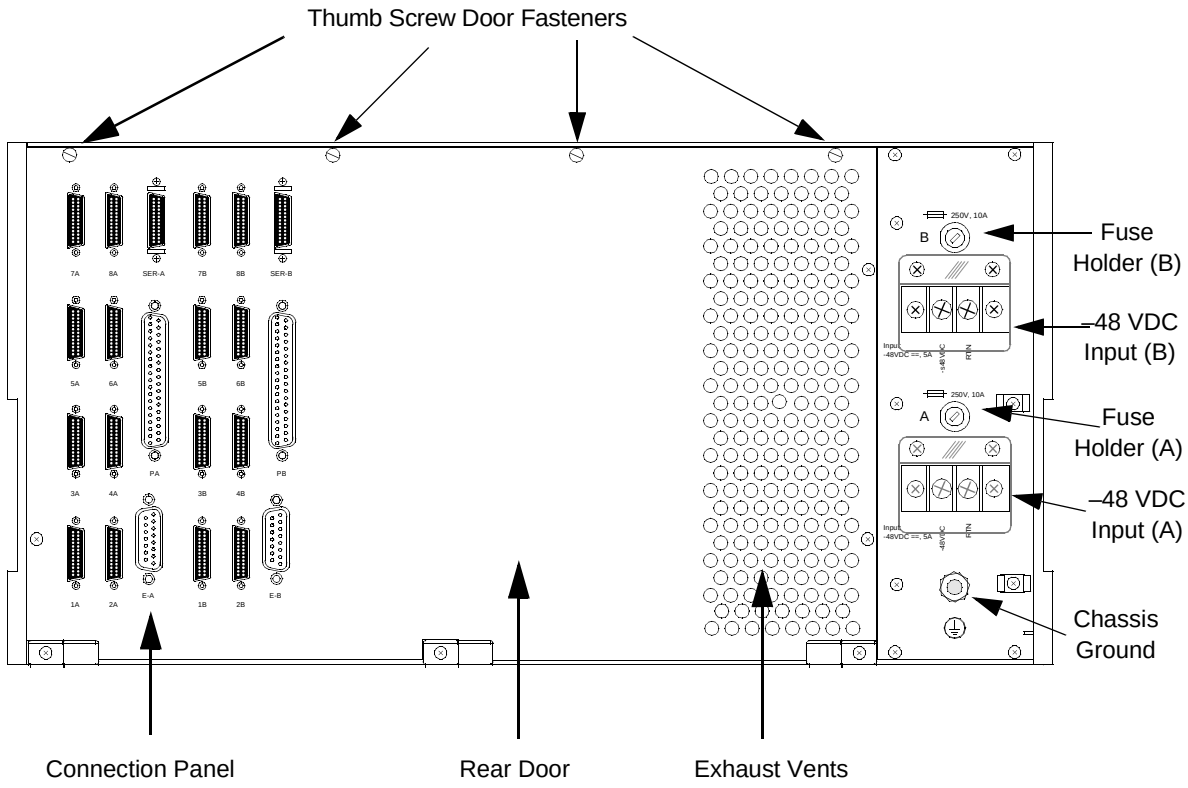


Figure 2.1: SS7 Shelf Back View (Door Closed)

2. Locate the P2 ribbon cable and connect the cable to the bottom, left side DIN connectors, labeled 1 and 3, as shown in Figure 2.2. *The red stripe (pin 1) goes toward the right.*

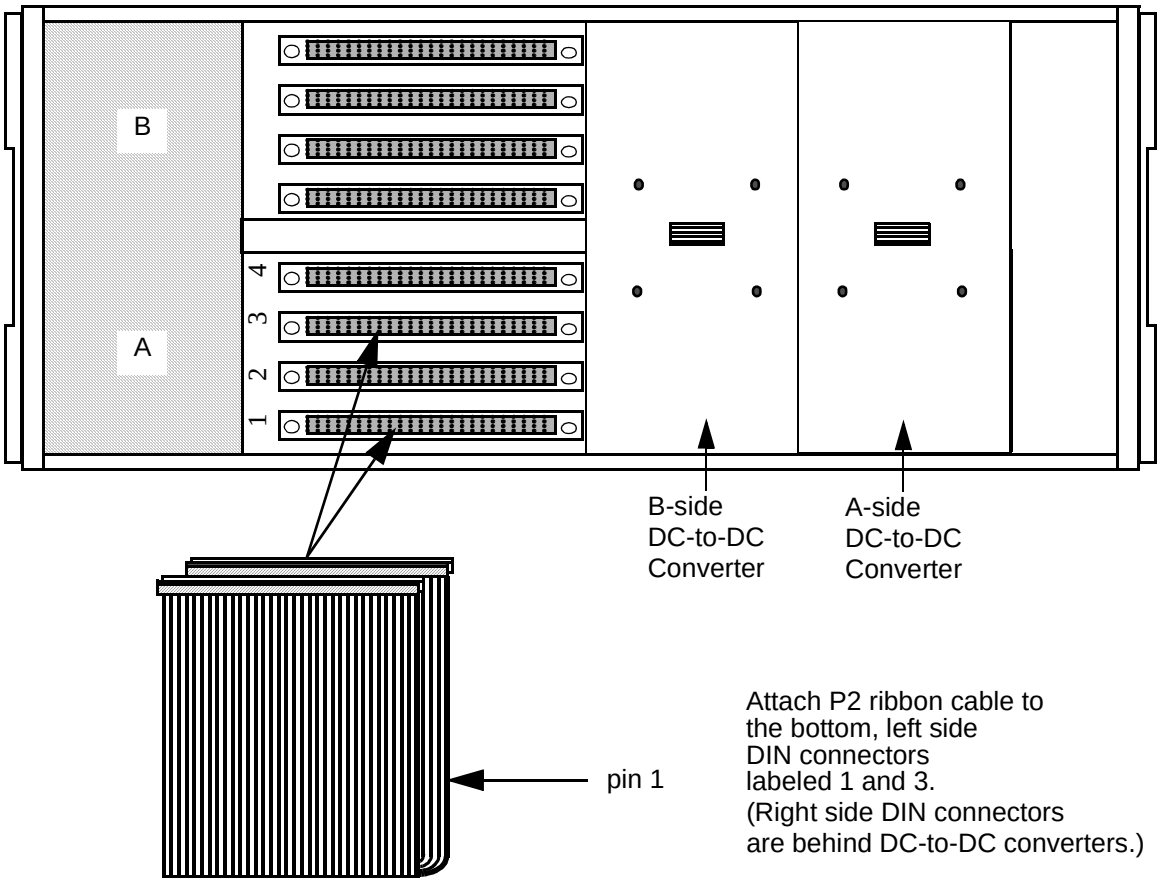


Figure 2.2: Connecting P2 Ribbon Cable

3. If you have a non-redundant configuration, go to Step 5.

   If you have a redundant configuration, locate the second P2 ribbon cable and connect the cable to the top, left side DIN connectors, labeled 1 and 3, as shown in Figure 2.3. *The red stripe (pin 1) goes toward the right.*

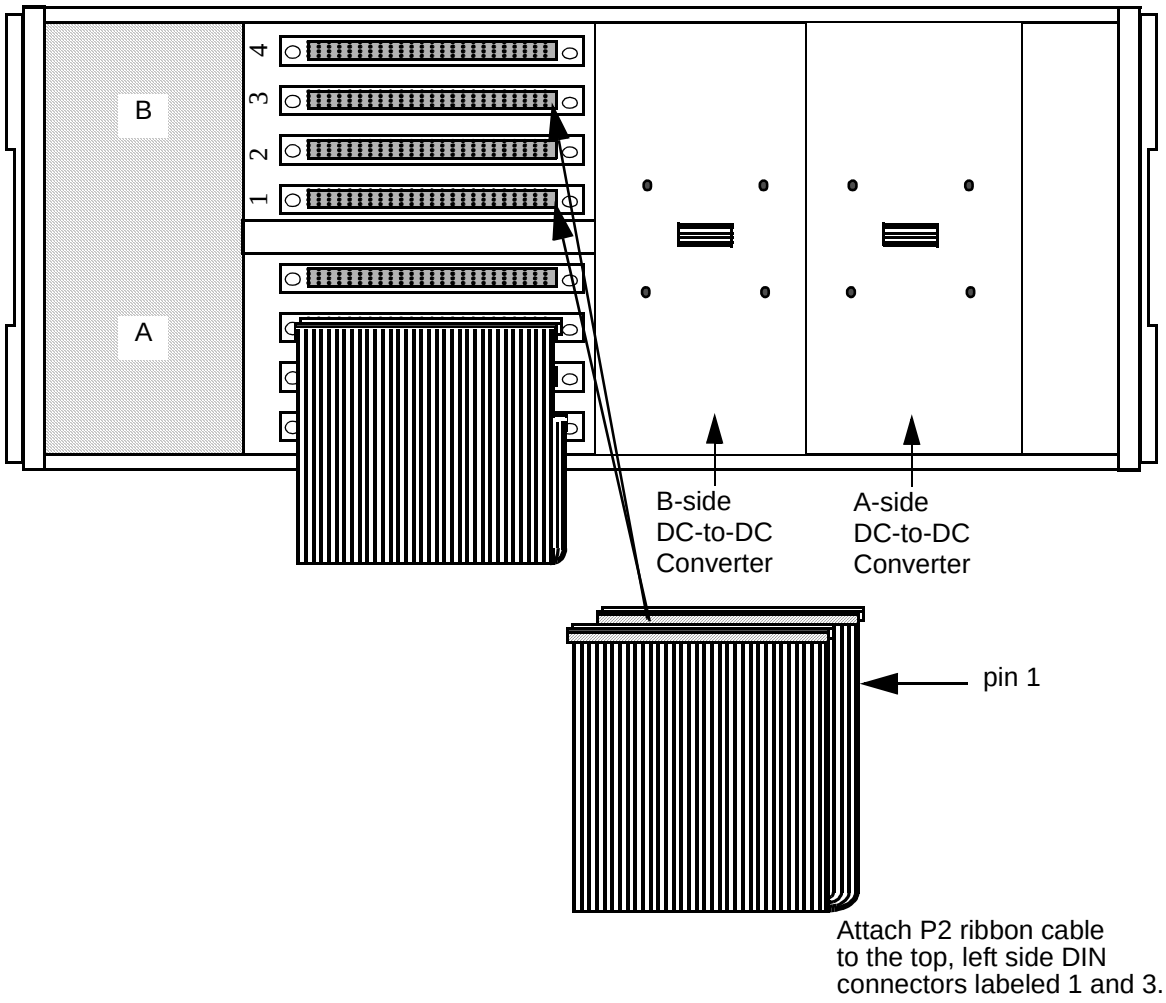


Figure 2.3: Connecting P2 Ribbon Cable

4. Close the rear door and hand tighten the top captive screws.

5. For NEBS-compliant earthquake rack installation, you *must* locate and install the brackets that were shipped with your VME SS7 shelf.

   To install these brackets, complete the following:

   a. Identify which bracket goes on which side. The right-side bracket is stamped with the part number 700-06753-01. The left-side bracket is stamped with the part number 700-06845-01.

   b. Locate the four M5 x 10mm SEM screws included in the bracket kit.

   c. Hold each bracket flush against the appropriate side of the SS7 VME shelf and line up the side screw holes; make sure the front bracket bend is parallel with the SS7 chassis rack-mount ears (refer to Figure 2.4).

   d. Use a #2 Phillips-head screwdriver to hand-tighten two screws on the side of each bracket.

   e. Use an adjustable torque wrench equipped with a #2 Phillips bit to tighten each screw to 21 to 29 in.-lb.

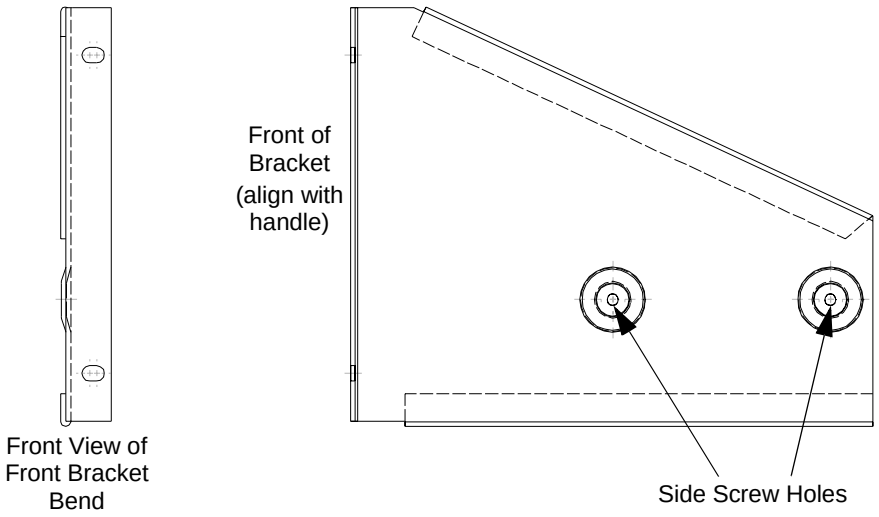


Figure 2.4: Right-side NEBS Bracket

6. Install the VME SS7 Shelf in a 19-inch rack.

   a. Make sure your screws go through both the shelf handles and the mounting brackets (if you chose to install them).

   b. Use an adjustable torque wrench to tighten all four appropriate mounting screws in accordance with common workmanship standards.

NOTE: The shelf requires its own –48 VDC power source, which is not supplied by Cisco Systems. The shelf cannot be powered by the VCO/4K.

NOTE: Cisco Systems recommends using at least two people to install the SS7 shelf.

7. Go to the front of the shelf and verify that both power switches are in the off position (down).

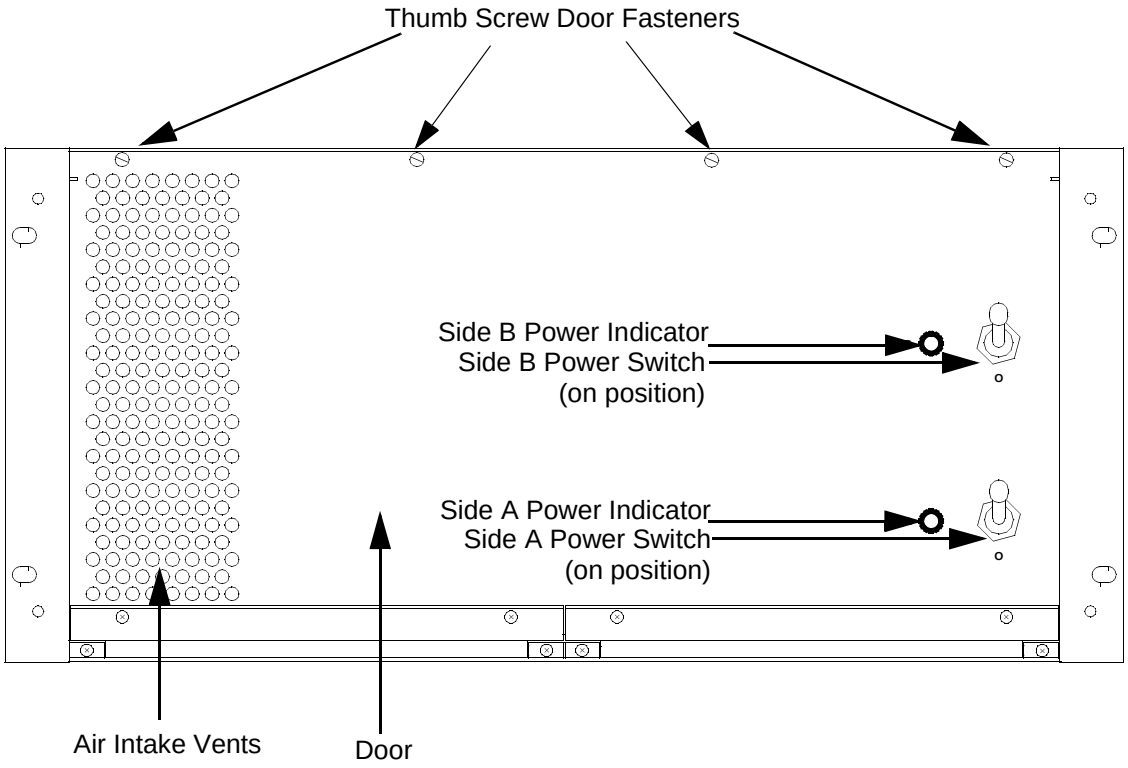


Figure 2.5: SS7 Shelf Front View (Door Closed)

The toggle switches are designed with spring-loaded locking mechanisms so the shelf cannot be accidentally powered on or off. Unlock the toggle switch by gently pulling the switch outward. Flip the switch downward to turn the power off. When the toggle is in the proper position, the spring mechanism locks the switch in place.

8. Attach building ground to the ground point on the rear of the SS7 chassis (see Figure 2.7). Use 14 AWG green/yellow stranded wire and an M5 lug. Attach the M5 lug to the SS7 chassis ground point according to the stackup shown below (Figure 2.6). Use an adjustable torque wrench to tighten the nut to 21 to 29 in.-lb.

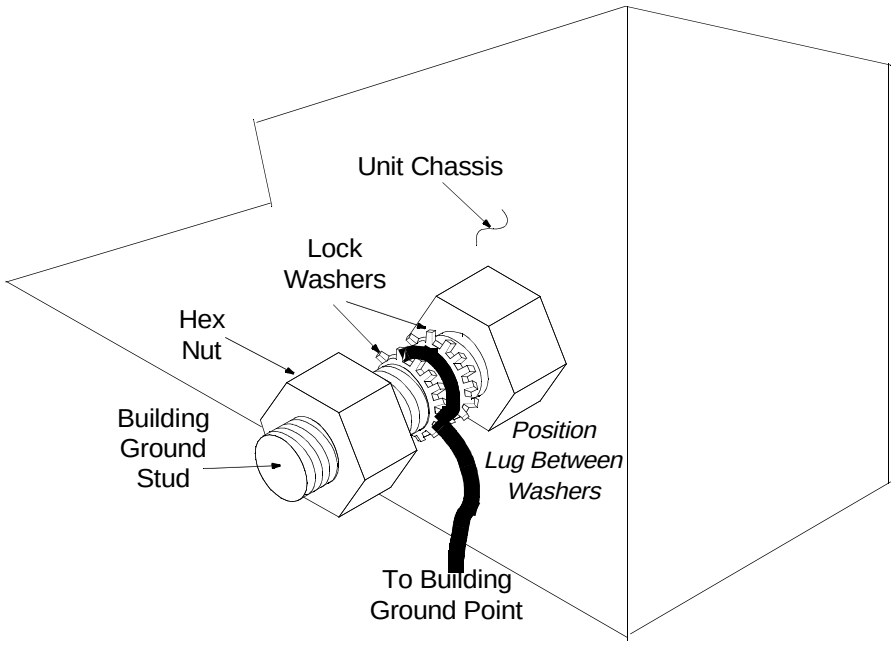


Figure 2.6: Ground Stud Stackup

9. Configure your –48 VDC power source (purchased separately) for the VME SS7 Shelf.

CAUTION: Disconnect the power source supplying the feed circuit(s) that enter your SS7 shelf. Tag the circuits at the disconnect points to warn others not to turn the power on while work is being completed. Leave the power feed disconnected until you have completed power source installation in the shelf and are ready to test it.

The power source connector on the SS7 shelf is a terminal block with an attached 3-pin modular pigtail power connector, as shown in Figure 2.7.

NOTE: All shelves, whether they are used in a non-redundant or redundant configuration, are equipped with two DC-to-DC converters, which are identical and interchangeable.

For redundant systems, the DC-to-DC converter for side A is mounted on the right, and the converter for side B is mounted on the left (when viewed from the rear of the unit). Each converter operates independent of the other; they do not share the load. If you have a problem with a converter on a redundant shelf, that side of the shelf will not operate until you replace the converter.

For non-redundant systems, the converter on side A supplies all of the DC voltages for shelf slots 1 through 4. The converter installed in side B can be used as a spare. In redundant systems, the converter in side B supplies power for shelf slots 5 through 8.

10. Make a power cable for your shelf. Use 14 AWG color-coded stranded wire per the pinouts represented in Figure 2.7. Cisco Systems supplies a male three-pin connector mate for the end of your power cable; white conductors are –48 VDC, A-side or B-side inputs, and black conductors are Battery Return, A-side or B-side inputs.

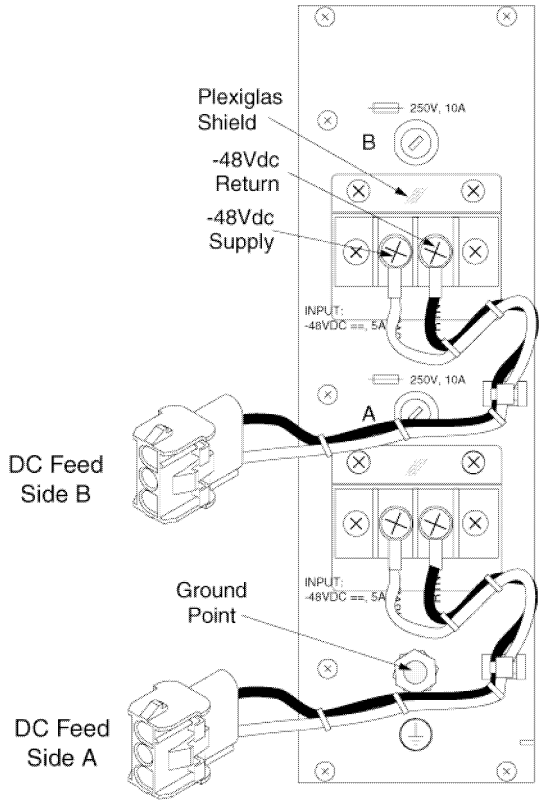


Figure 2.7: VME SS7 Shelf A&B Power Inputs

11. Go to the back of the shelf and connect the power cable(s) to the DC feed connector(s) shown in Figure 2.7 (A for a non-redundant system or A and B for redundant systems).
12. Connect the other end of the power cable(s) to your power source.
13. Reconnect the power source supplying the feed circuit(s) that enter your SS7 shelf.
14. To test the shelf, go to the front of the shelf and flip the power switch(es) on (upward position) and verify that the appropriate green LED illuminates.

    If the LED does not illuminate, perform the following checks:

    a. Verify that the –48 VDC power source is on and that there is –48 VDC at the end of the cable to the shelf.
    b. Check the main power fuse (one each on A and B inputs).
    c. **Turn the power off** and remove the DC-to-DC power converter back panel. Verify that the connectors for the DC-to-DC converter are tight.
    d. Check the 15-amp fuse on the converter.
    e. Verify that the green LED on the front of the system is good.

    If you cannot locate the problem, contact Cisco Systems Technical Support.

15. After you have tested the shelf, toggle the shelf's power switch(es) off (downward position).

2.5 INSTALLING CARDS AND LINKS: NON-REDUNDANT OR SIDE A

To install the SS7 subsystem hardware on a non-redundant system or side A of redundant systems, complete the following steps:

CAUTION: Observe antistatic precautions when handling the cards to avoid damaging sensitive CMOS devices. Wear a ground strap connected to the grounded system equipment frame whenever dealing with system components.

1. If you haven't already done so, verify that you have all of the hardware components with your order. If you are missing components, contact Cisco Systems Technical Support.
2. Power off the VCO/4K, according to the instructions in the *VCO/4K System Maintenance Manual*.
3. Remove the SPARC CPU/Sbus assembly from its packaging.

   Verify that the DIP switches are set correctly, as shown in Figure 2.9.

   CAUTION: If the DIP switch settings are not correct on the SPARC CPU5V, the VCO/4K database gets corrupted. This applies to both redundant and non-redundant systems.

   Figure 2.8 shows the location of the switches on a SPARC CPU5V.

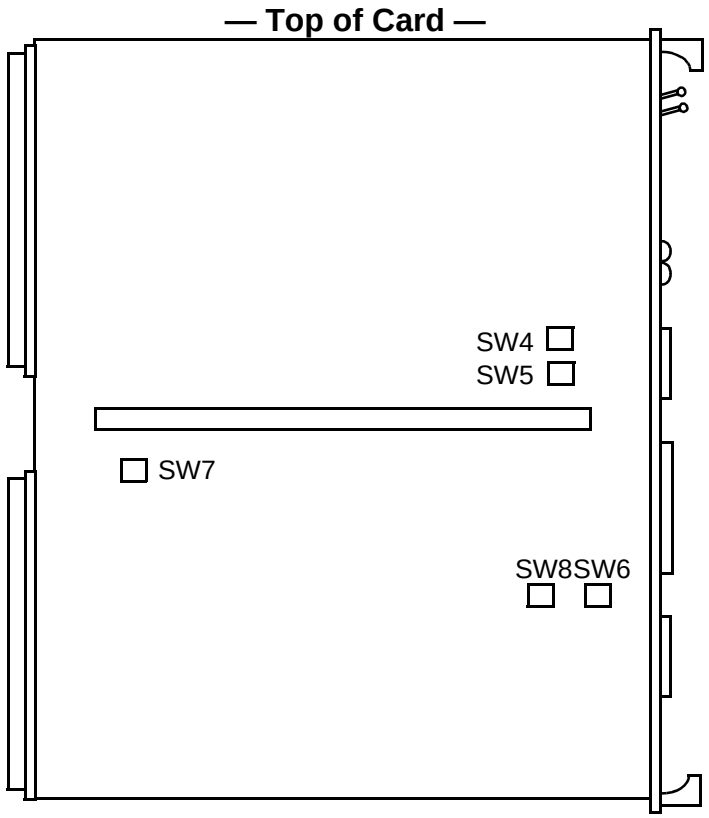
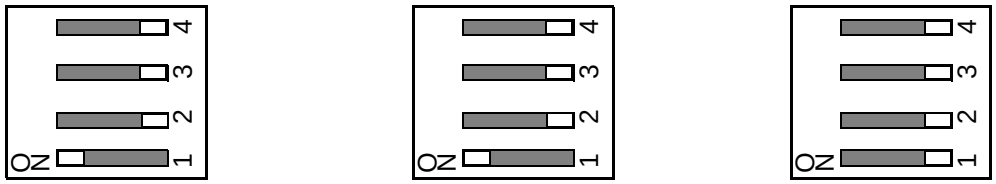


Figure 2.8: SPARC CPU5V Switch Locations

Figure 2.9 shows the correct setting for each switch on a SPARC CPU5V.



SW4

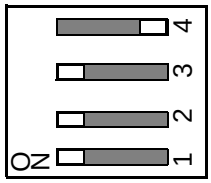
S4-1 = on
S4-2 = off
S4-3 = off
S4-4 = off

SW5

S5-1 = on
S5-2 = off
S5-3 = off
S5-4 = off

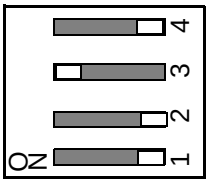
SW6

S6-1 = off
S6-2 = off
S6-3 = off
S6-4 = off



SW7

S7-1 = on
S7-2 = on
S7-3 = on for production systems
off for development systems
S7-4 = off



SW8

S8-1 = off
S8-2 = off
S8-3 = on
S8-4 = off

NOTE: When S7-3 is on, the SPARC-5V resets when the VCO/4K reboots.

Figure 2.9: SPARC CPU5V Switch Settings

4. Install the SPARC CPU/Sbus assembly and Storage Subsystem. These components are installed in the SS7 Shelf, as shown in Figure 2.10.

   Note that the cards are installed horizontally. Rotate the CPU/Sbus assembly 90° counterclockwise, and install it in slots 1 and 2. Rotate the Storage Subassembly 90° counterclockwise, and install it in slots and 3 and 4.

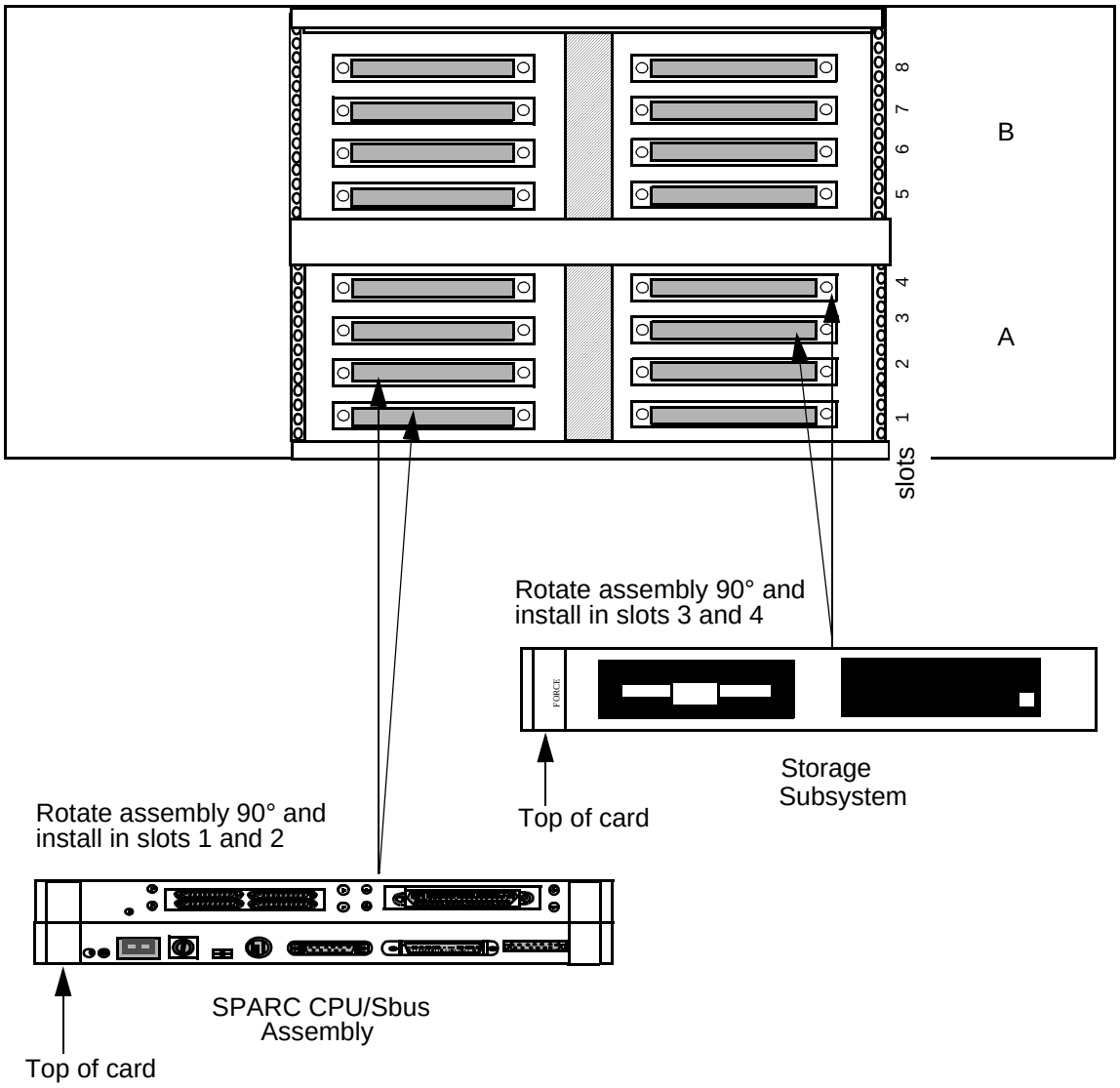


Figure 2.10: Installation of SPARC CPU/Sbus Assembly and Storage Subsystem: VME SS7 Shelf

NOTE: When attaching cables, make sure you tighten all jack screws and other locking mechanisms to secure cables in place.

NOTE: Cable installation instructions in this manual apply to non-SVX environments only.

5. Locate the LAN A cable (D-15 male) on the bottom right inside wall of your shelf (see Figure 2.11).
6. Connect the D-15 male connector to the Ethernet port of your SPARC CPU (see Figure 2.12).
7. Connect and lock into place a shielded Ethernet AUI cable or Ethernet transceiver to the D-15 female locking connector labeled E-A on the back distribution panel, as shown in Figure 2.13.
8. Connect to your LAN.

   *NOTE: You **must** use shielded cables for your Ethernet AUI connections.*

   NOTE: The SS7 subsystem, VCO/4K, and host(s) must all be on the same IP subnet.

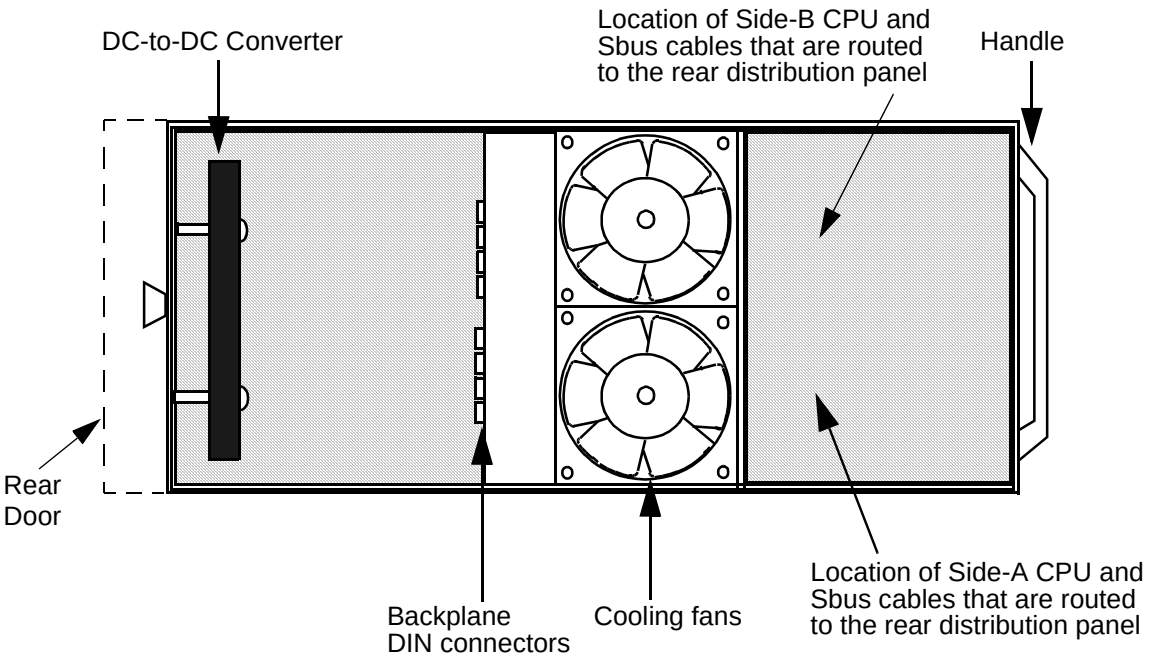


Figure 2.11: Cutaway Side View

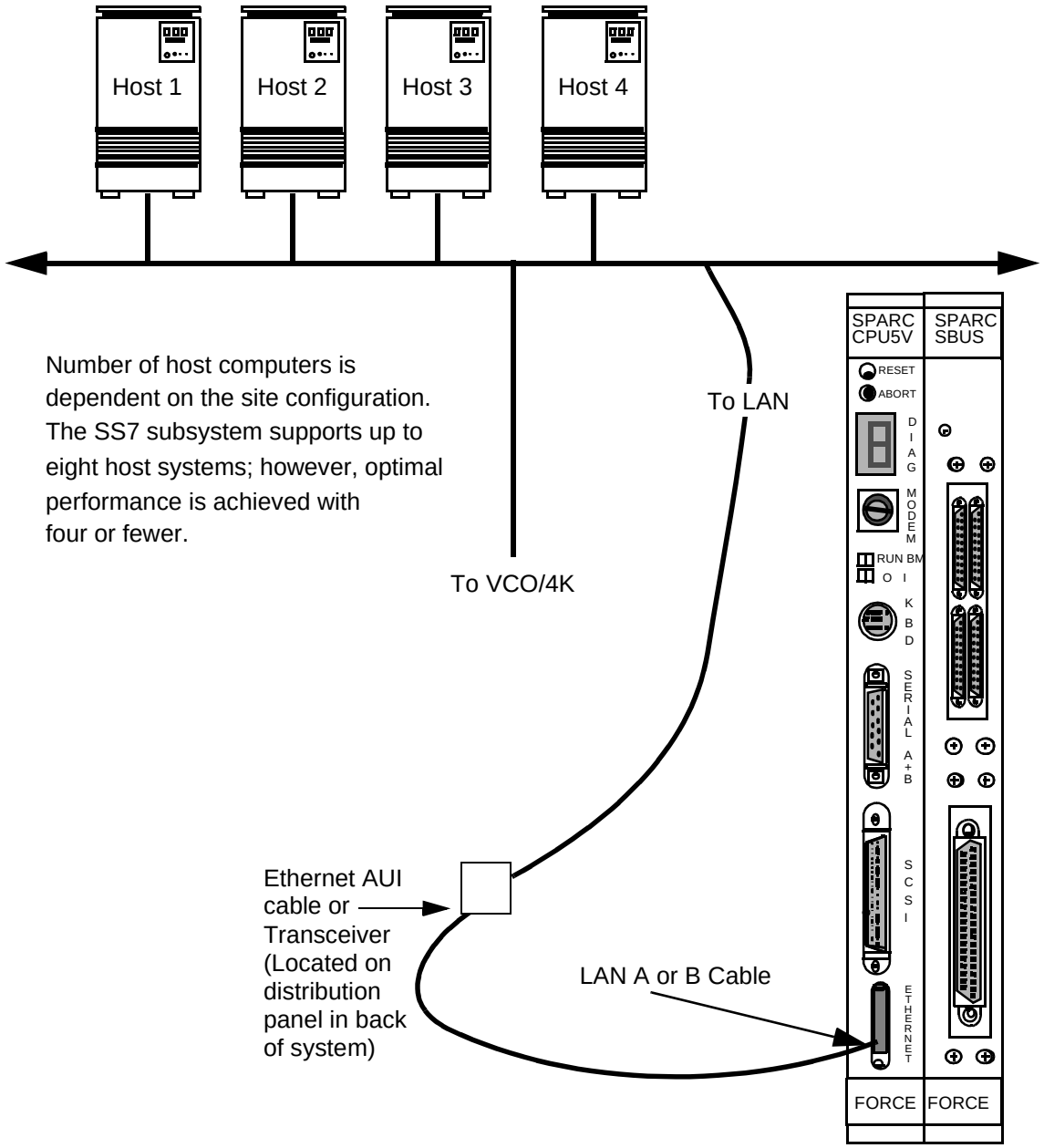


Figure 2.12: Ethernet Pathway

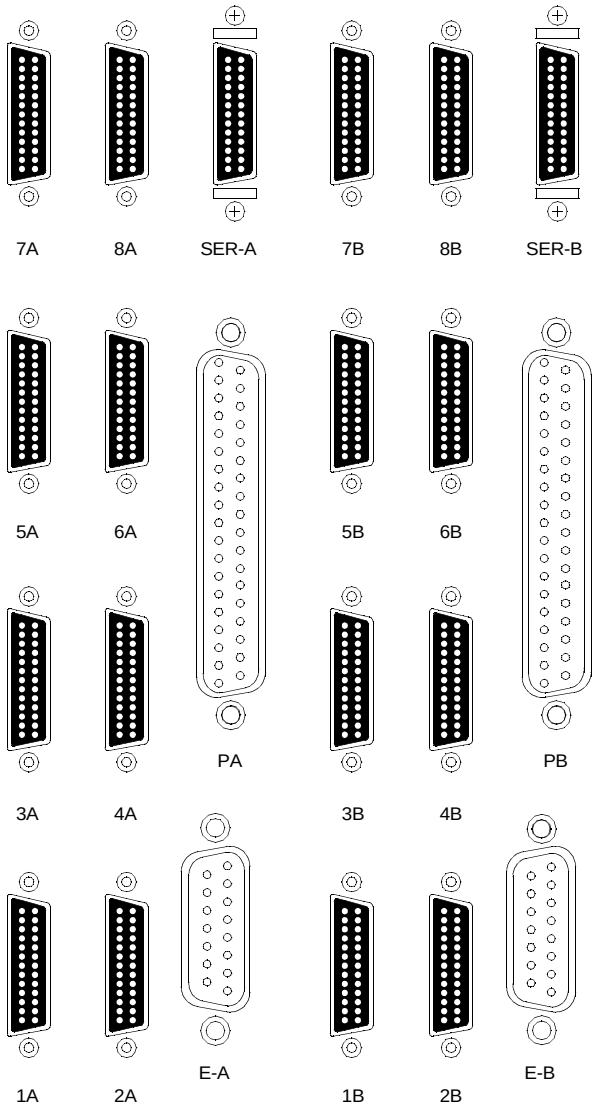


Figure 2.13: SS7 Shelf Rear Distribution Panel

9. Locate the A-side SS7 link ribbon cables (micro D-26 male) located in the internal ribbon-cable bundle on the bottom right inside wall of your shelf. The cables should be labeled 1A through 4A, then 5A through 8A for 8 links.

   Connect the micro D-26 male end of the first cable (1A) to port 1 on the Sbus card. The associated micro D-26 female end of the cable located on the rear distribution panel (refer to Figure 2.13) is labeled identically.

NOTE: If your site has only one link to the SS7 network, it must connect to the Sbus Port 1.

If your site has more than one SS7 network link, connect the micro D-26 male end of the additional cables to their respective ports (ports 2 to 4 or 2 to 8) on the Sbus card, as shown in Figure 2.14 or Figure 2.15. The micro D-26 female ends located on the rear distribution panel (refer to Figure 2.13) are labeled identically (2A through 8A).

10. Locate the SS7 link cables (micro D-26 male to DB-37 male) and connect the micro D-26 end(s) of the cable(s) to the appropriate location(s) on the rear distribution panel. Connect the DB-37 end(s) of the cable(s) to the link(s) for your SS7 network.

    NOTE: If your site has more than one link, pay close attention to which physical Sbus port location you are using for each SS7 network link. This information is required to configure the system later in the installation.

    Cisco Systems strongly recommends clearly labeling both ends of each cable with the physical port location (1A to 8A).

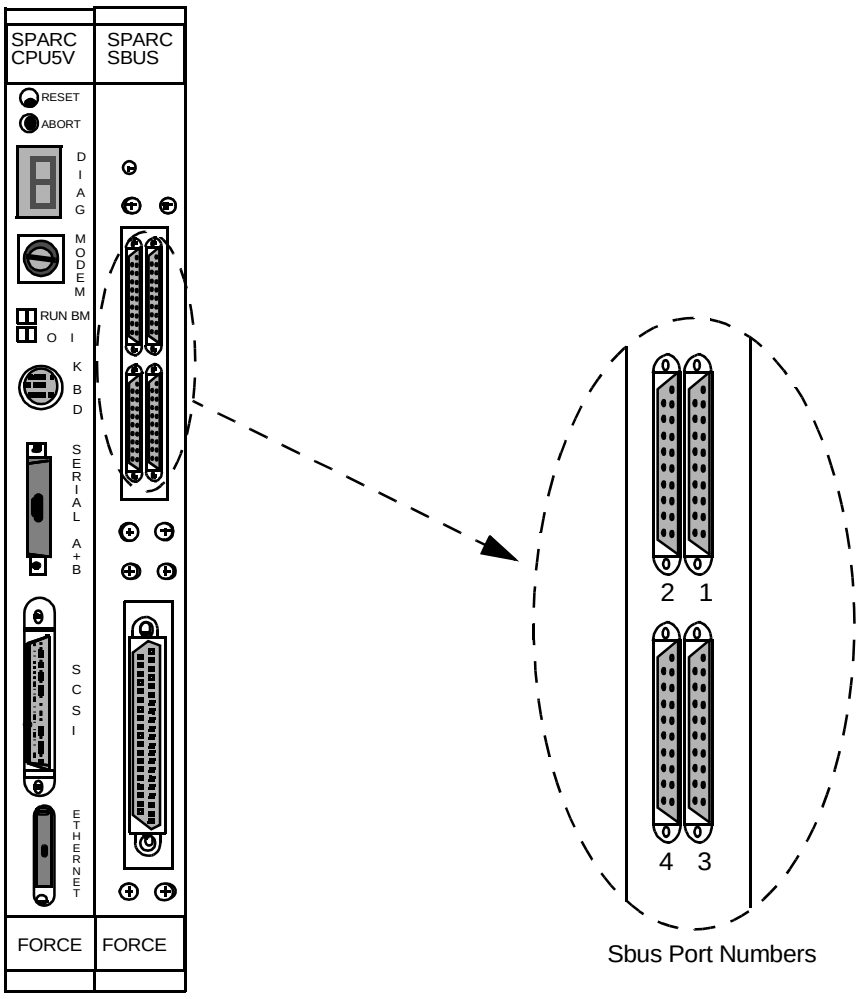


Figure 2.14: SS7 Network Link Connections (4-link)

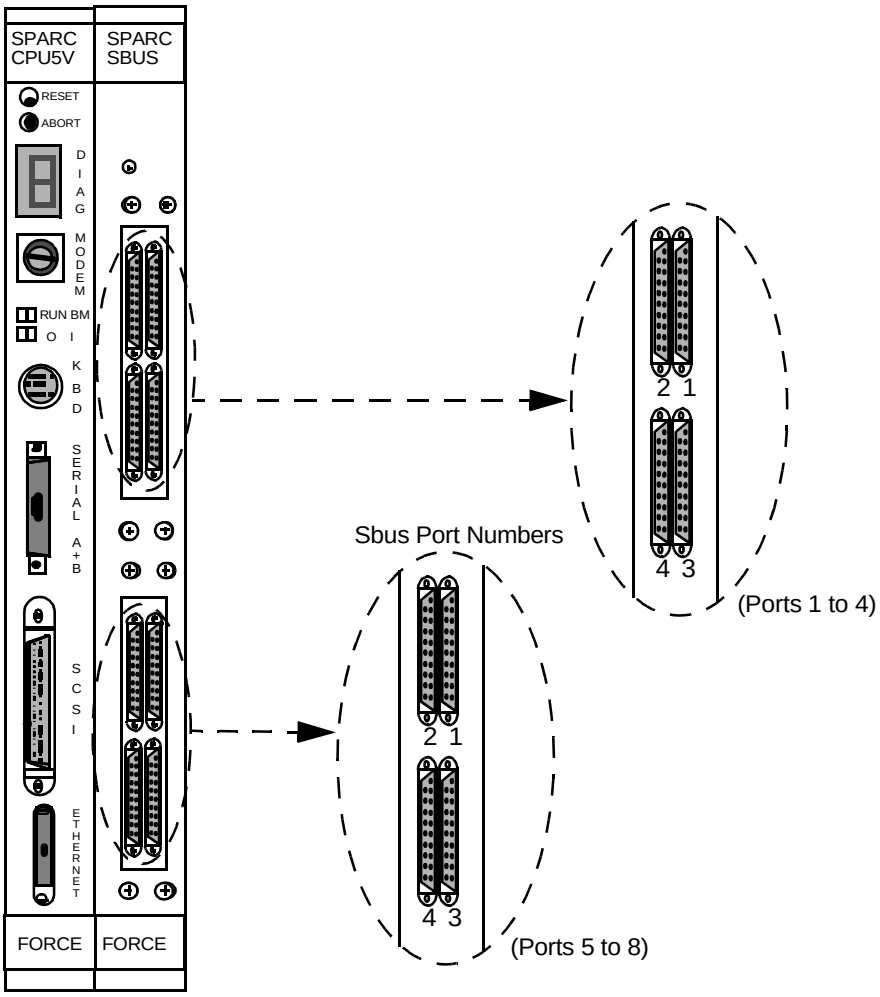


Figure 2.15: SS7 Network Link Connections (8-link)

2.6 INSTALLING PERIPHERALS: NON-REDUNDANT SYSTEMS

This subsection describes how to install the peripherals on non-redundant systems. If you have a redundant system, skip this section.

To install the SS7 subsystem peripherals on non-redundant systems, complete the following steps:

NOTE: When attaching cables, make sure you tighten all jack screws and other locking mechanisms to secure cables in place.

1. Locate the Serial A ribbon cable (micro D-26 male) located in the internal ribbon-cable bundle on the bottom right inside wall of your shelf.
2. Connect the micro D-26 male end of the cable to the SPARC CPU port labeled Serial A+B, as shown in Figure 2.16.

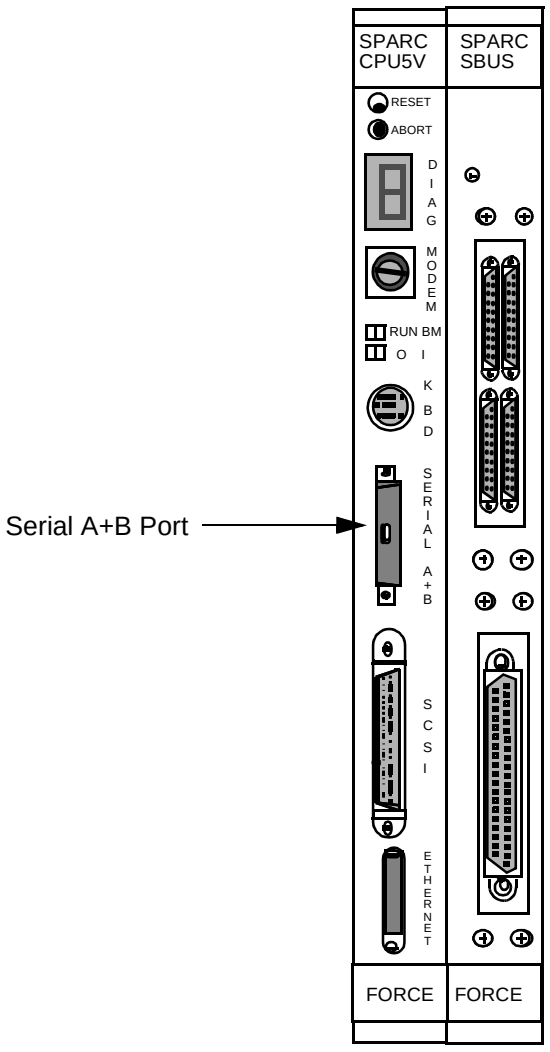


Figure 2.16: Serial A+B Port Location

3. Locate the two-way splitter cable:
   a. Connect the single DB-25 end to the micro D-26 female connection labeled SER-A on the rear distribution panel.
   b. Connect the DB-25 end labeled A to the ASCII terminal via the null cable or null modem, as shown in Figure 2.17.
   c. Leave the end labeled B disconnected; it is used in redundant configurations.

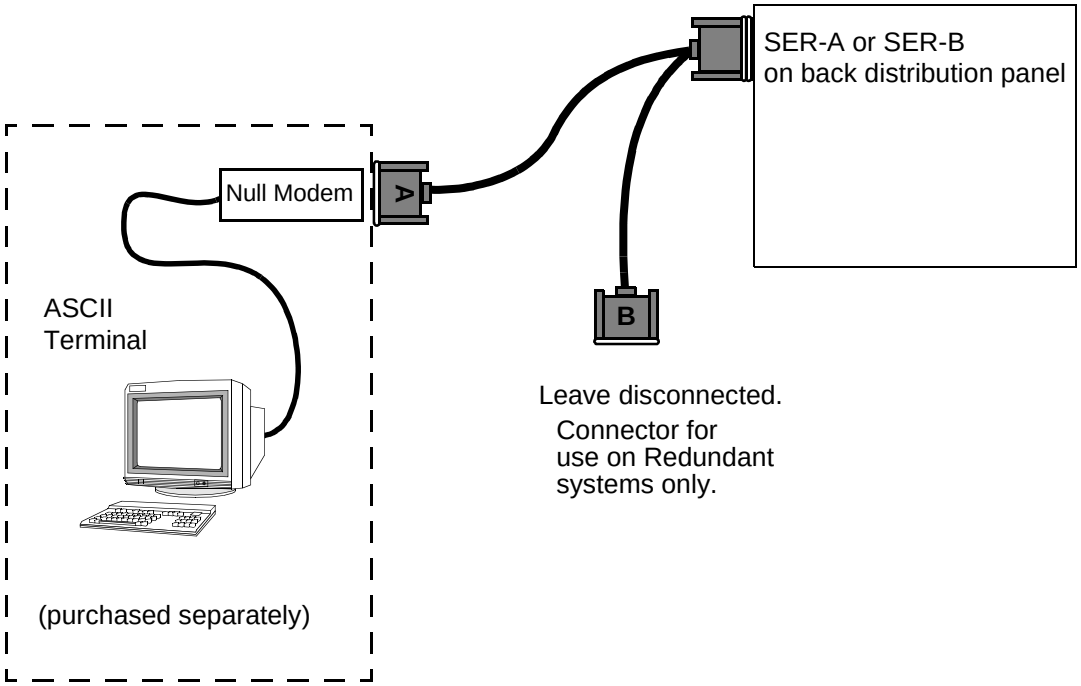


Figure 2.17: ASCII Terminal Connections

NOTE: When configuring the ASCII terminal, use VT100 Emulation. Also, use data settings of 9600 baud, eight bit, no-parity, and one stop bit.

CAUTION: Attaching a terminal to the SS7 SPARC serial port can cause an SS7 reboot/halt under certain failure conditions. This complication is in the Solaris operating system, which hangs when the VT console is either powered down or has a power supply failure.

Cisco Systems recommends disconnecting the VT terminal from the SPARC processor serial connection and either (a) reconnecting for active use only, or (b) leaving disconnected and performing all terminal functions via a Telnet session from the host or another connectivity device.

NOTE: ***Step 4 and Step 5 are unnecessary for 8-link systems.*** *Instead, you must arrange dial-up to a local network and Telnet access to the SS7 subsystem.*

4. Locate and set up the modem according to the instructions in the manufacturer's user documentation.
5. To connect the modem to the system:
   a. Locate the ribbon cable labeled PA (DB-37) located in the internal ribbon-cable bundle on the bottom right inside wall of your shelf.
   b. Connect the end of the cable to the serial port on the Sbus.
   c. Locate and connect the DB-37 end of the three-way splitter cable to the DB-37 connector labeled PA on the rear distribution panel (see Figure 2.13).
   d. Locate the 25-foot EIA/TIA-232 cable.
   e. Locate the TTY0 connector on the three-way splitter. Connect the DB-25 female end of the EIA/TIA-232 cable to TTY0 (see Figure 2.18).
   f. Connect the DB-25 male end of the 25-foot EIA/TIA-232 cable to the modem.

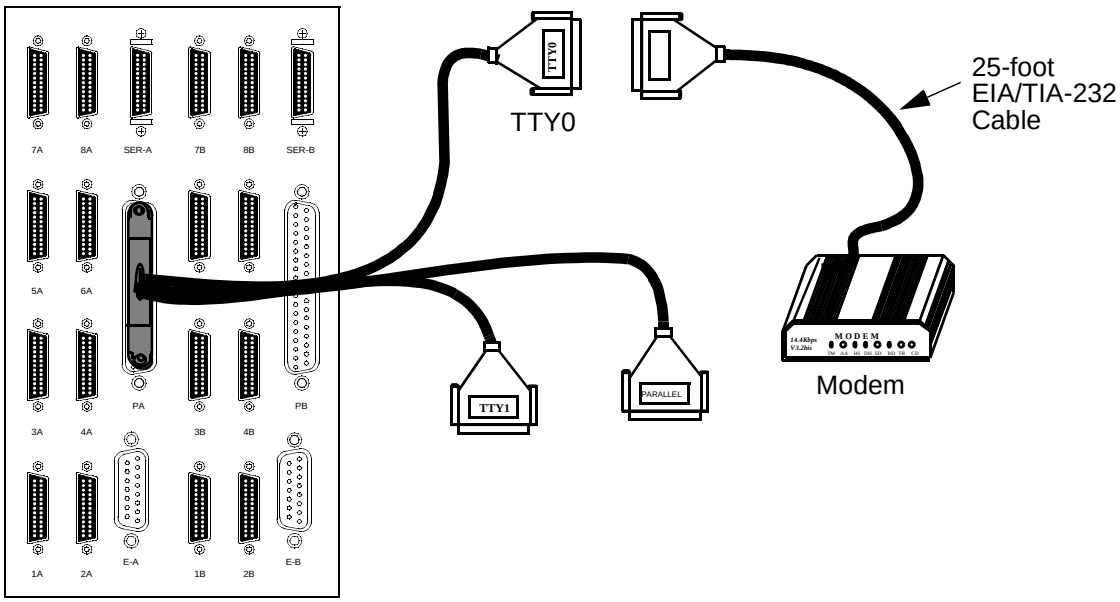


Figure 2.18: Modem Drive Connections

NOTE: ***Step 6 is optional.*** *The CD-ROM drive is only needed to re-install Solaris.*

6. To connect the CD-ROM drive to the system:
   a. Locate the CD-ROM drive's SCSI cable.
   b. Connect one end of the SCSI cable to the SCSI port on the SPARC CPU (see Figure 2.19).
   c. Connect the other end of the SCSI cable to the CD-ROM drive.
   d. Set the SCSI address of the CD-ROM drive to a value of 6.

NOTE: The CD is not hot pluggable. The SS7 subsystem must be re-booted before you can use the drive.

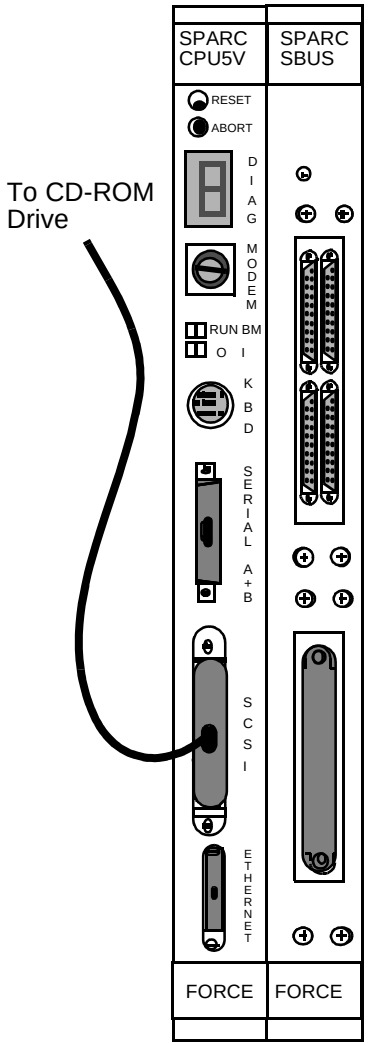


Figure 2.19: CD-ROM Drive Connections

7. Power up the VCO/4K according to the instructions in the *VCO/4K System Maintenance Manual*.
8. Power up the shelf by flipping the A-side power switch on (upward). See Figure 2.5 for the location of the power switch.

This completes the hardware installation process for non-redundant systems.

NOTE: Make sure you remove the CD-ROM and its associated cable after completing your software download process. Close and resecure the front door after CD-ROM removal. The front door must be closed in order to meet NEBS and EMI compliance.

2.7 INSTALLING CARDS AND LINKS: SIDE B OF REDUNDANT SYSTEMS

This section describes how to install the cards on side B of redundant systems.

CAUTION: Observe antistatic precautions when handling the cards to avoid damaging sensitive CMOS devices. Wear a ground strap connected to the grounded system equipment frame whenever dealing with system components.

1. Remove the second SPARC CPU/Sbus assembly from its packaging.

   Verify that the DIP switches are set correctly.

   CAUTION: If the DIP switch settings are not correct on the SPARC CPU5V, the VCO/4K database gets corrupted. This applies to both redundant and non-redundant systems.

   Figure 2.8 shows the location of the switches on a SPARC CPU5V. Figure 2.9 shows the correct setting for each switch.

2. Install the second SPARC CPU assembly and Storage Subsystem into side B of the SS7 shelf as shown in Figure 2.20.

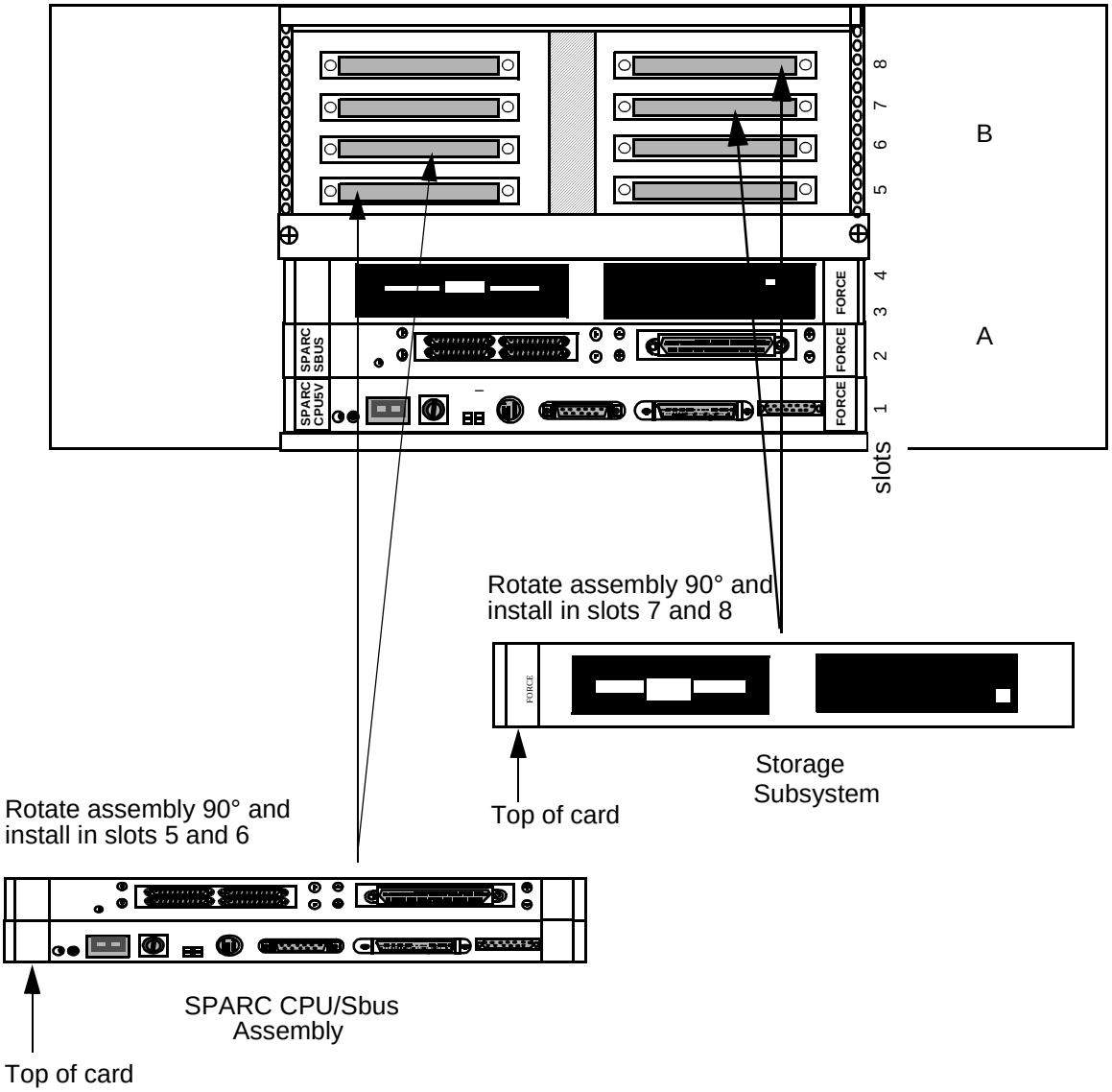


Figure 2.20: Assembly and Storage Subsystem Installation: Side B VME SS7 Shelf

NOTE: When attaching cables, make sure you tighten all jack screws and other locking mechanisms to secure cables in place.

NOTE: Cable installation instructions in this manual apply to non-SVX environments only.

3. Locate the LAN B cable (D-15 male) on the top right inside wall of your shelf.
4. Connect the D-15 male connector to the Ethernet port of your B-side (top) SPARC CPU (see Figure 2.12).

5. Connect and lock into place a shielded Ethernet AUI cable or Ethernet transceiver to the D-15 female locking connector labeled E-B on the back distribution panel, as shown in Figure 2.13.
6. Connect to your LAN.

   *NOTE: You **must** use shielded cables for your Ethernet AUI connections.*

   NOTE: The SS7 subsystem, VCO/4K, and host(s) must all be on the same IP subnet.
7. Locate the B-side SS7 link ribbon cables (micro D-26 male) located in the internal ribbon-cable bundle on the top right inside wall of your shelf. The cables should be labeled 1B through 4B, then 5B through 8B for 8 links.

   Connect the micro D-26 male end of the first cable (1B) to port 1 on the Sbus card. The associated micro D-26 female end of the cable located on the rear distribution panel (refer to Figure 2.13) is labeled identically.

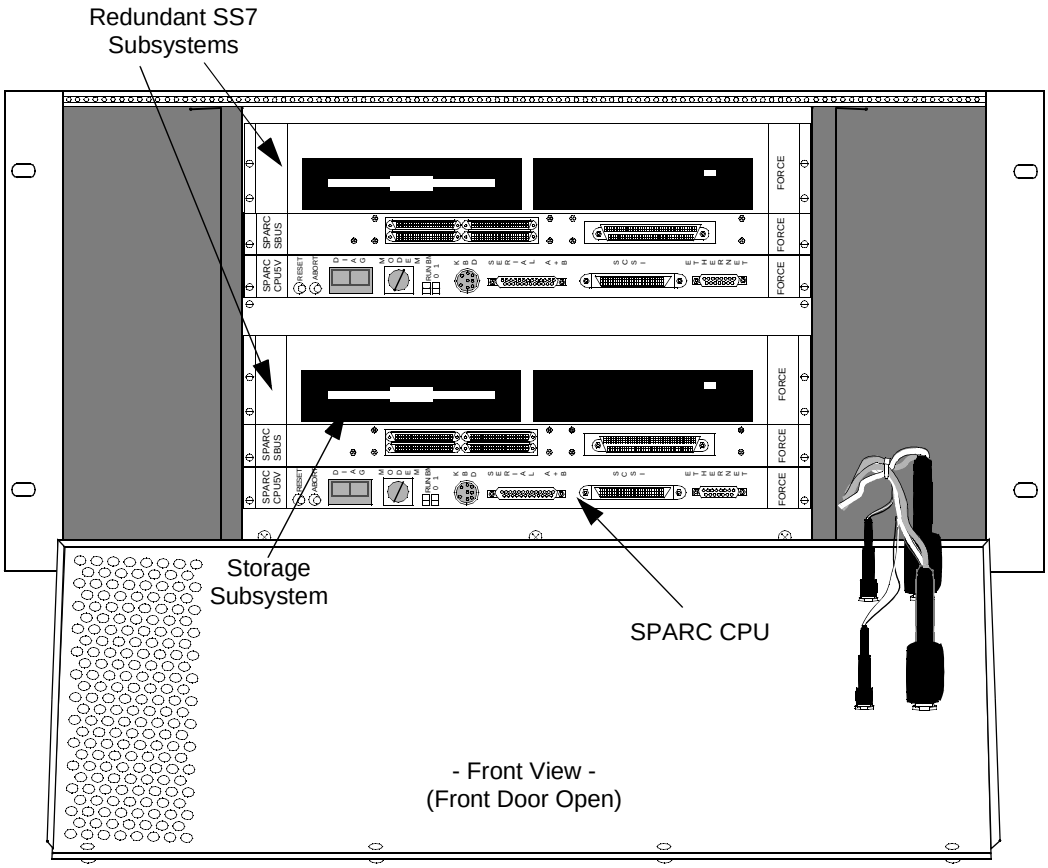
   NOTE: If your site has only one link to the SS7 network, it must connect to the Sbus Port 1.

   If your site has more than one SS7 network link, connect the micro D-26 male end of the additional cables to their respective ports (ports 2 to 4 or 2 to 8) on the Sbus card, as shown in Figure 2.14 or Figure 2.15. The micro D-26 female ends located on the rear distribution panel (refer to Figure 2.13) are labeled identically (2B through 8B).
8. Locate the SS7 link cables (micro D-26 male to DB-37 male) and connect the micro D-26 end(s) of the cable(s) to the appropriate location(s) on the rear distribution panel. Connect the DB-37 end(s) of the cable(s) to the back of your SS7 selector switch, as described in *Section 2.9*.

   NOTE: If your site has more than one link, pay close attention to which physical Sbus port location you are using for each SS7 network link. This information is required to configure the system later in the installation.

   Cisco Systems strongly recommends clearly labeling both ends of each cable with the physical port location (1B to 8B).

Figure 2.21 shows the front view of a redundant SS7 system with the door open.



NOTE: Cable harness is left out for visual clarity.

Figure 2.21: Front View of Redundant SS7 System

2.8 INSTALLING PERIPHERALS: REDUNDANT SYSTEMS

This subsection describes how to install the peripherals on redundant systems. First, follow the directions in *Section 2.6* to install the SS7 subsystem peripherals on Side A. Then, complete the following steps:

NOTE: When attaching cables, make sure you tighten all jack screws and other locking mechanisms to secure cables in place.

1. Locate the Serial B ribbon cable (micro D-26 male) located in the internal ribbon-cable bundle on the top right inside wall of your shelf.
2. Connect the micro D-26 male end of the cable to the B-side SPARC CPU (top) port labeled Serial A+B, as shown in Figure 2.16.
3. Locate the two-way splitter cable:
   a. Connect the single DB-25 end to the micro D-26 female connection labeled SER-B on the rear distribution panel.
   b. Connect the DB-25 end labeled A to the ASCII terminal via the null cable or null modem, as shown in Figure 2.17.
   c. Leave the end labeled B disconnected.

NOTE: When configuring the ASCII terminal, use VT100 Emulation. Also, use data settings of 9600 baud, eight bit, no-parity, and one stop bit.

CAUTION: Attaching a terminal to the SS7 SPARC serial port can cause an SS7 reboot/halt under certain failure conditions. This complication is in the Solaris operating system, which hangs when the VT console is either powered down or has a power supply failure.

Cisco Systems recommends disconnecting the VT terminal(s) from the SPARC processor serial connection and either (a) reconnecting for active use only, or (b) leaving disconnected and performing all terminal functions via a Telnet session from the host or another connectivity device.

NOTE: ***Step 4 and Step 5 are unnecessary for 8-link systems.*** *Instead, you must arrange dial-up to a local network and Telnet access to the SS7 subsystem.*

4. Set up the modem and A/B transfer switch following the instructions in the manufacturers' documentation.

   NOTE: Cisco Systems recommends a separate modem to each side. Otherwise, the modem connection can hang periodically and be unusable. Follow directions from Section 2.6 for connecting directly to a modem.

5. To connect the modem to the redundant SS7 subsystem (see Figure 2.22):
   a. Remove the EIA/TIA-232 cable from the three-way splitter cable off of the PA connector on the rear distribution panel.
   b. Connect the TTY0 connector of the three-way splitter to the side A input at the back of the SS7 selector switch.
   c. Locate the ribbon cable labeled PB (DB-37) located in the internal ribbon-cable bundle on the top right inside wall of your shelf.
   d. Connect the D-37 female end of the cable to the serial port on the Side-B Sbus.
   e. Locate and connect the DB-37 end of the three-way splitter cable to the DB-37 connector labeled PB on the rear distribution panel (see Figure 2.13).
   f. Connect the TTY0 connector of the three-way splitter to the side B input at the back of the A/B transfer switch.
   g. Locate the 25-foot EIA/TIA-232 cable.
   h. Connect the DB-25 end of the cable to the outgoing port of the A/B transfer switch.
   i. Connect the EIA/TIA-232 end of the cable to the modem.

   NOTE: ***Step 6 is optional.*** *The CD-ROM drive is only needed to re-install Solaris.*

6. Connect the CD-ROM drive to either side of the SS7 subsystem by completing the following steps:
   a. Locate the CD-ROM drive's SCSI cable.
   b. Connect one end of the SCSI cable to the SCSI port on the SPARC CPU (see Figure 2.19).
   c. Connect the other end of the SCSI cable to the CD-ROM drive.
   d. Set the SCSI address of the CD-ROM drive to a value of 6.

   NOTE: The CD is not hot pluggable. The SS7 subsystem must be re-booted before you can use the drive.

   NOTE: Make sure you remove the CD-ROM and its associated cable after completing your software download process. Close and resecure the front door after CD-ROM removal. The front door must be closed in order to meet NEBS and EMI compliance.

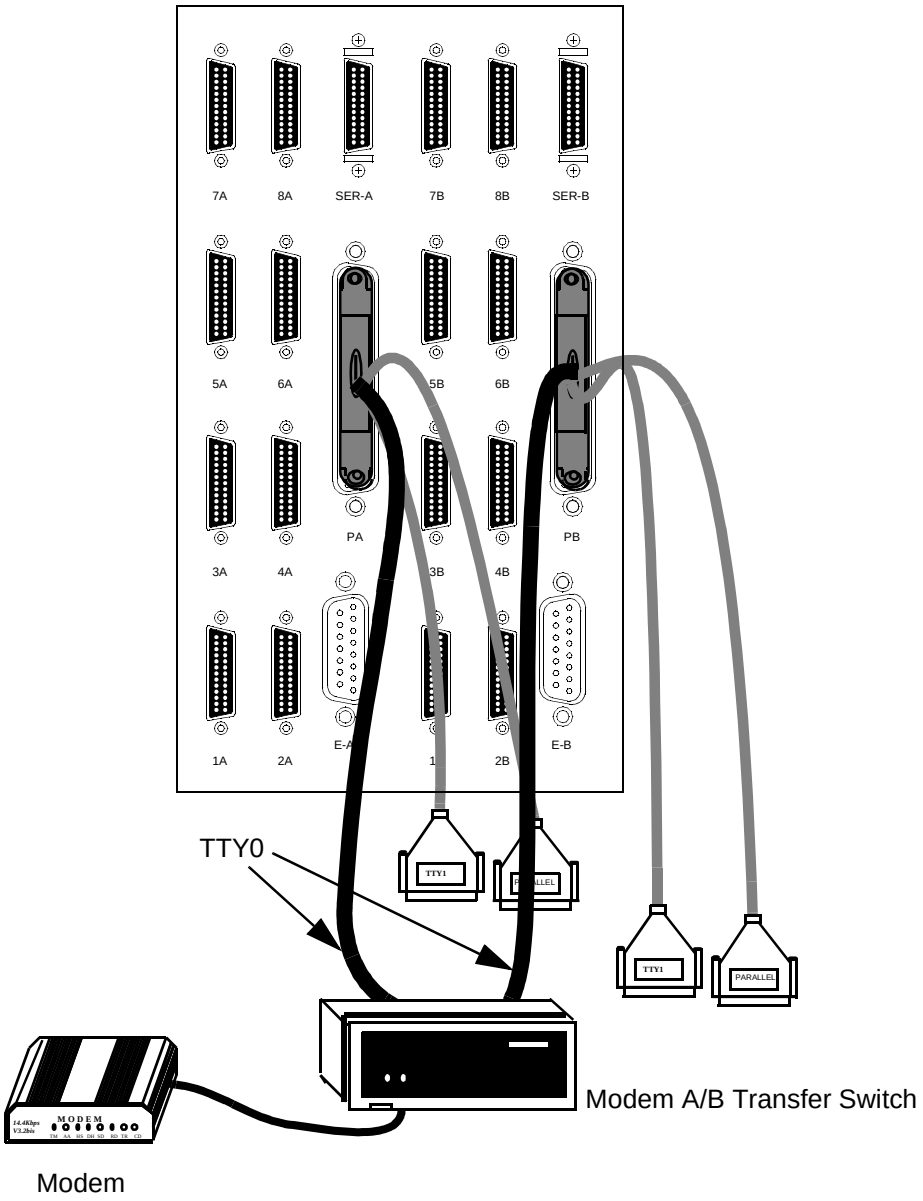


Figure 2.22: Modem Connections on Redundant Systems

2.9 INSTALLING SS7 SELECTOR SWITCH: REDUNDANT SYSTEMS

This subsection describes how to install the SS7 selector switch, which provides SS7 link redundancy connection to both side A and side B of the redundant configuration. The selector switch has redundant power supplies (see Figure 2.23).

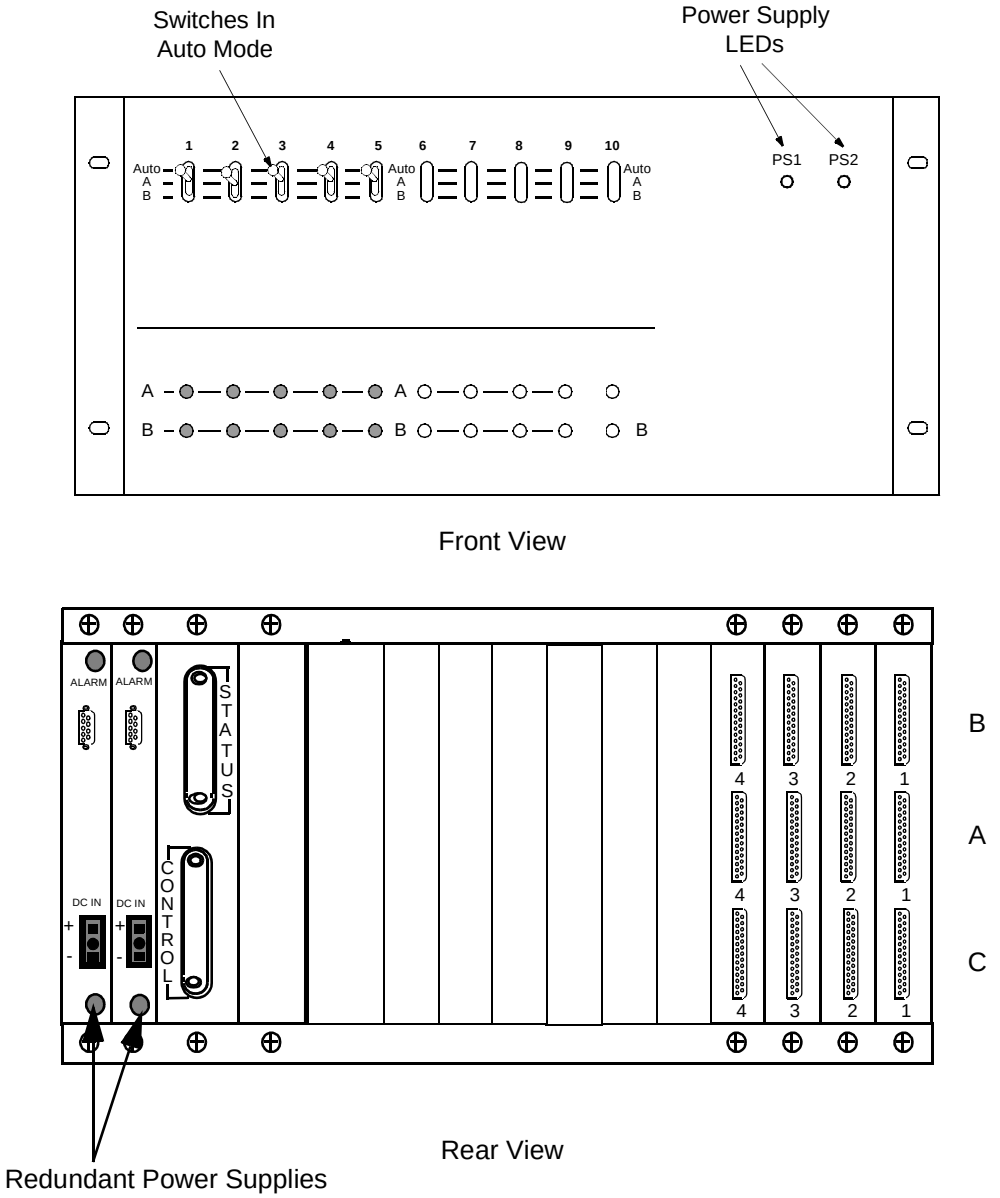


Figure 2.23: SS7 Selector Switch (4-link)

2.9.1 Specifications

Part Number:	Contact your Cisco Systems sales representative
Dimensions:	Height — 8.75 inches (22.23 cm) Depth — 8 inches (20.32 cm) Width —19 inches (48.26 cm) Designed to mount in a 19” rack
Power Input:	–48 VDC — dual input
Compliance:	EMI/EMC FCC Part 15 (U.S. and Canada) EN55022/50082 (for Europe) NEBS GR-63-CORE (Issue 1, 1995) GR-1089-CORE (Issue 2, 1997) UL 1950 CSA C22.2 EN60950 (for Europe) IEC-950
Weight (max.):	8 lbs.
Operational Temperature Range:	40 to 100°F 10 to 40°C
Operational Temperature Gradient:	15°F (10°C) per hour
Operational Humidity:	20 to 80 percent (%), non-condensing
Operational Altitude:	0 to 10,000 ft 0 to 3,048 m

2.9.2 Installation

The selector switch is designed to mount in a 19-inch utility rack, –48 VDC device, which is separate from the VCO/4K. When installing in a site with AC power only, an AC-to-DC rectifier of sufficient DC output power is required.

CAUTION: Do not power the SS7 selector switch by the VCO/4K.

The required power for the empty chassis is 2 watts (maximum draw). Each board draws a maximum of 5 watts. For example, running 4 link cards in a chassis draws 20 watts, *plus* 2 watts for the chassis requirements, which equals a *total* of 22 watts.

The SPARC CPU/Sbus assembly must be properly installed in each side of the VCO/4K’s control subrack before you begin to install the SS7 selector switch.

To install the SS7 selector switch, complete the following steps:

1. Install the SS7 selector switch in the utility rack.

   NOTE: In addition to these components, the installation requires a separate 19-inch utility rack with its own –48 VDC source. These components must be purchased separately. If your site has AC power only, an AC-to-DC rectifier of sufficient DC output power must be used.

   CAUTION: There is no power switch on the SS7 selector switch. The unit will power up as soon as input DC voltage is applied.

2. Attach building ground to the ground point (see Figure 2.26) on the rear of the SS7 selector switch. Use 14 AWG green/yellow stranded wire and an M4 lug. Attach the M4 lug to the SS7 selector switch ground point according to the stackup shown below (Figure). Use an adjustable torque wrench to tighten the nut to 13 to 18 in.-lb.

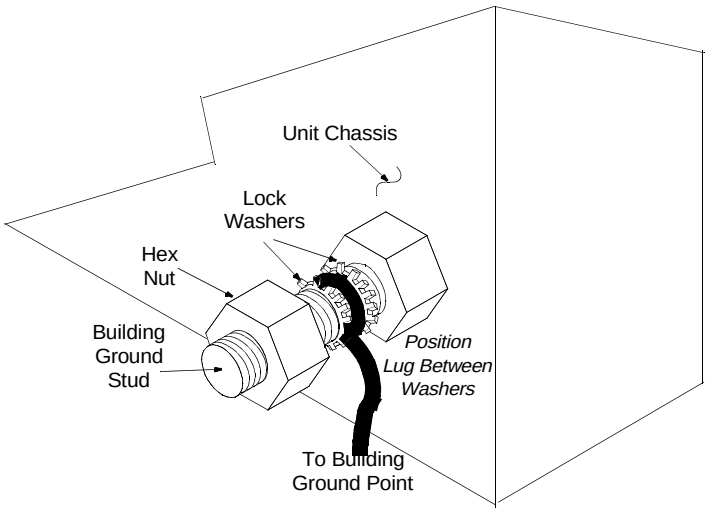


Figure 2.24: Ground Stud Stackup

3. Configure the –48 VDC power source (purchased separately) for the SS7 selector switch.

   CAUTION: Disconnect the power supplying the feed circuit(s) that enter your SS7 selector switch. Tag the circuits at the disconnect points to warn others not to turn the power on while work is being completed. Leave the power feed disconnected until you have completed power source installation in the selector switch and are ready to test it.

4. Make two power cables for the SS7 selector switch.

   NOTE: All selector switches are equipped with two DC-to-DC converters, which are identical and interchangeable. Due to current sharing, either converter can support a fully loaded system.

   Cisco Systems supplies male three-pin connector mates for the end of your power cables. Use 14 AWG color-coded stranded wire. One wire is for –48 VDC, one wire is for –48 VDC return, and the final wire is a green/yellow ground wire, as shown below in Figure 2.25.

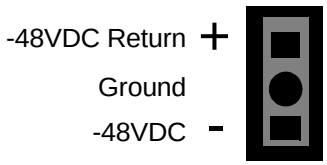


Figure 2.25: SS7 Selector Switch DC Power Inlet

5. Locate the Status/Control cable assembly.
   a. Connect the Status/Control connectors to the rear of the SS7 selector switch as shown in Figure 2.26.
   b. Connect the DB-25 end of the Status/Control cable (marked A side) to the Serial B connector off of the SER-A connector on the rear distribution panel of the SS7 subsystem (see Figure 2.26).
   c. Connect the DB-25 end of the Status/Control cable (marked B side) to the Serial B connector off of the SER-B connector on the rear distribution panel of the SS7 subsystem (see Figure 2.26).

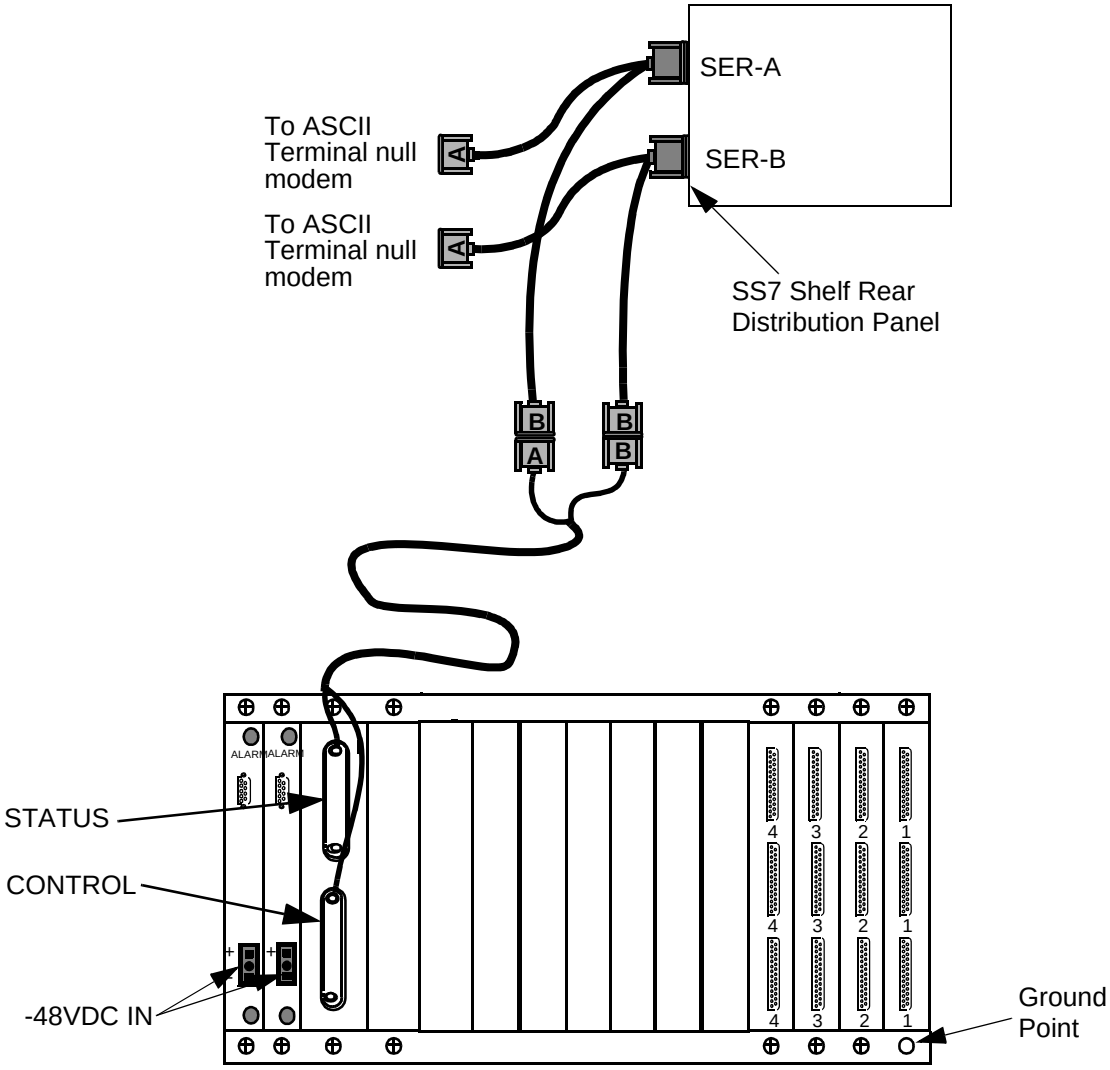


Figure 2.26: SS7 Selector Switch Status/Control Connections

6. Locate the DB-25 male to DB-37 female cables and connect the SS7 subsystem sides A and B to the SS7 selector switch as follows:
   a. From the rear of the SS7 selector switch, connect the DB-25 male end of the first cable in position A-1, as shown in Figure 2.27.
   b. Connect the DB-37 end of the first cable to the DB-37 end of the cable connected to the 1A connector off of the rear distribution panel of the SS7 subsystem.

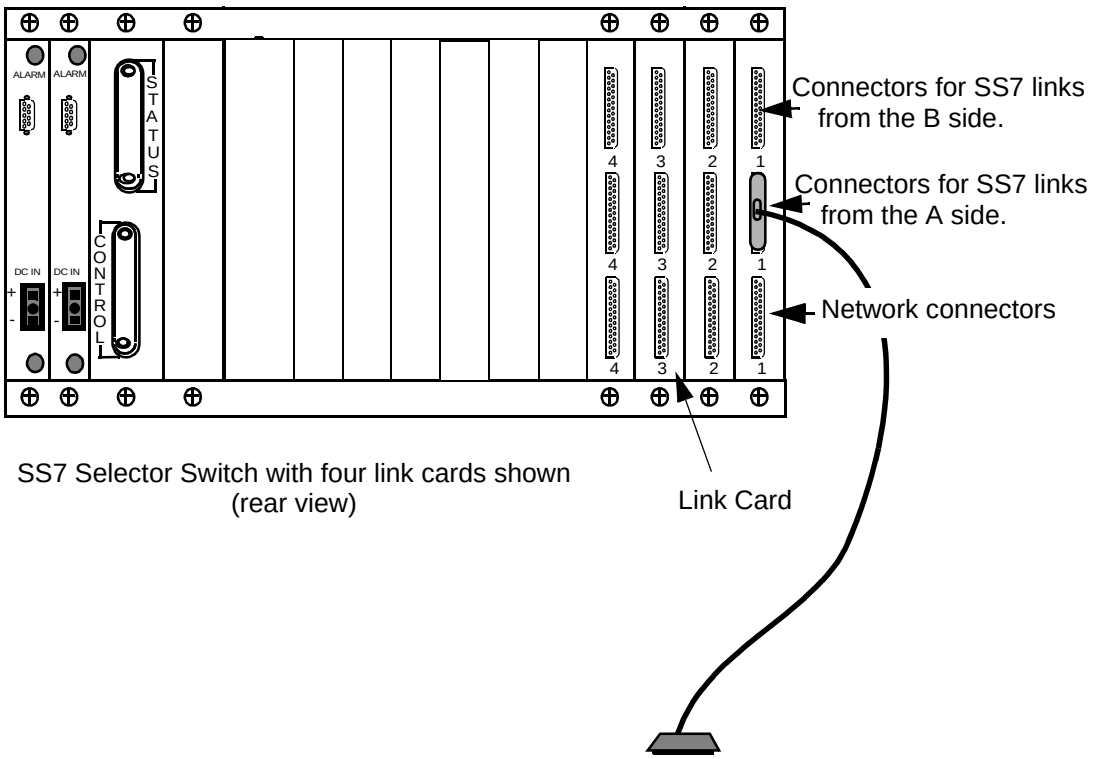


Figure 2.27: SS7 Selector Switch to Signaling Links

   c. Repeat step a and step b until all SS7 selector switch signaling links are connected on side A (2A through 4A and 5A through 8A).
   d. Connect the DB-25 male end of the next cable to position B-1 (see Figure 2.27).
   e. Connect the DB-37 end of the fifth (or ninth) cable to the DB-37 end of the cable connected to the 1B connector off of the rear distribution panel of the SS7 subsystem.
   f. Repeat steps d and e until all SS7 selector switch signaling links are connected on side B (2B through 4B and 5B through 8B).
   g. Locate the DB-25 male to DB-37 male cables.
   h. Connect the DB-25 male end of the first cable to position C-1 of the SS7 selector switch EIA/TIA-449 connectors labeled C (See Figure 2.27). Connect the DB-37 male end to your SS7 network.
   i. Repeat step h until all of the SS7 selector switch signaling links labeled C are connected to your SS7 network.

7. Go to the front of the SS7 selector switch and put the A/B toggle switch in the AUTO position, as shown in Figure 2.28, for each link connected.

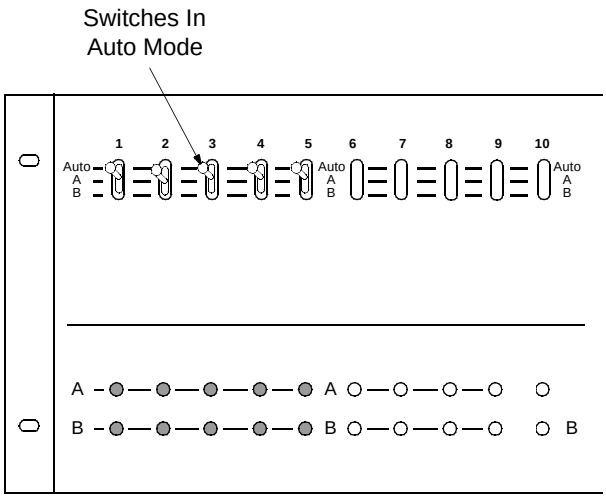


Figure 2.28: SS7 Selector Switch A/B Toggle Switch Position

NOTE: Keep the A/B toggle switch in the AUTO position during normal operation. When performing maintenance on the links, this switch can be used to lock the SS7 selector switch on side A or side B after you initiate a switchover as described in Section 5.6. Do not use the A/B toggle switch to initiate a switchover.

8. Reconnect the power supplying the feed circuit(s) that enter your SS7 selector switch.
9. Toggle the SS7 VME shelf A-side and B-side power switches to the on position (up).

   NOTE: Remember, the SS7 selector switch powered on when input DC voltage was applied.

10. Power up the VCO/4K, following the instructions in the *VCO/4K System Maintenance Manual*.

SECTION 3
SOFTWARE INSTALLATION AND NETWORK CONFIGURATION

3.1 INTRODUCTION

The software is loaded on the SS7 subsystem hard drive at the factory. However, when you boot your system for the very first time, the **sys-config** script runs automatically to configure the system on the network. The **sys-config** script performs the following tasks:

- Adds the SS7 subsystem's hostname and IP address to the **/etc/hosts** file
- Sets the hostname in **/etc/rc.boot**
- Sets the domain name in **/etc/rc.single**
- Sets the **/usr/lib/zoneinfo/localtime** file
- Enables NIS if Network Information Services are requested

After **sys-config** executes, you must edit the SS7 subsystem's **/etc/hosts** file to include the hostnames and IP addresses of the systems connected to the SS7 subsystem. These systems include the VCO/4K, hosts and, if you have a redundant configuration, the second SS7 subsystem.

3.1.1 Network Information Worksheets

Before you boot the system, make copies of the following Network Information worksheets and fill them out with the assistance of your network administrator. If you have any questions, contact Cisco Systems Technical Support for assistance.

> **CAUTION: If any of this information is entered incorrectly, the system will not boot. *Solaris is case sensitive.***

Save the worksheets after you have completed the instructions in this section. You will need this information to configure the ISUP and TCAP software components as described in *Section 4.4*.

Network Information Worksheet 1

Non-Redundant Configurations

SS7 subsystem hostname _____
SS7 subsystem IP address _____
VCO/4K hostname _____
VCO/4K IP address _____
VCO/4K socket number _____

Redundant Configurations

SIDE A	SIDE B
SS7 hostname* _____	SS7 hostname* _____
SS7 IP address _____	SS7 IP address _____
VCO/4K hostname _____	VCO/4K hostname _____
VCO/4K IP address _____	VCO/4K IP address _____
VCO/4K socket number _____	VCO/4K socket number _____

All Configurations*f*

Name service _____
Domain name _____
Name server hostname _____
Name server IP address _____
Netmask _____
Root password _____
Is the SS7 subsystem part of a subnet**? ___

* These names must be the same except for the last character. The last character must be “a” for side A and “b” for side B.

f Use the same values for both sides of redundant configurations.

** In non-redundant configurations, we recommend the SS7 subsystem, VCO/4K, and host all be on the same IP subnet.

In a redundant configuration, *Cisco Systems recommends* the SS7 subsystem, VCO/4K, and host for side A be on the same IP subnetwork. The same is true for the systems on side B. The two sides can be on separate subnetworks, but the SS7 subsystem sides must be able to communicate with each other over the Ethernet. The side A host and side B host must be capable of routing Ethernet packets between the two subnetworks.

Network Information Worksheet 2

Application Host Information[f]

Host 1 hostname _____
Host 1 IP address _____
Host 1 socket number _____
Host 2 hostname _____
Host 2 IP address _____
Host 2 socket number _____
Host 3 hostname _____
Host 3 IP address _____
Host 3 socket number _____
Host 4 hostname _____
Host 4 IP address _____
Host 4 socket number _____
Host 5 hostname _____
Host 5 IP address _____
Host 5 socket number _____
Host 6 hostname _____
Host 6 IP address _____
Host 6 socket number _____
Host 7 hostname _____
Host 7 IP address _____
Host 7 socket number _____
Host 8 hostname _____
Host 8 IP address _____
Host 8 socket number _____

f Use the same values for both sides of redundant configurations.

3.1.2 Special Considerations For On-Line Help

The script has on-line Help, which you can access at any time by pressing **F6**. When the on-line Help first comes up, the following message appears:

```
If you choose F3 to change your entries, you will go back only
to the screens shown in the summary. To change previously-
confirmed information, you must halt and reboot your system
(i.e., boot).
```

Do not use the reboot method to change information after it has been confirmed. In some cases, the system will loop back to the middle of the installation script and you will be unable to reenter the desired data.

To correct or change information after it has been confirmed, complete the following steps:

1. Finish running the installation script using the default values.
2. Reboot the system.
3. Log in as **root**.
4. Change directory to **/usr/sbin**.
5. Enter the following command and press **Return:**

   sys-unconfig

   The following message appears:

   ```
   This program will unconfigure your system. It will cause it
   to revert to a “blank” system - it will not have a name or know
   about other systems or networks

   This program will also halt the system.

   Do you want to continue (y/n)?
   ```

6. Enter **y** and press **Return**. The system halts.
7. At the **ok** prompt, enter **boot** and press **Return**.

3.2 INSTALLING TCAP SOFTWARE

The TCAP software is purchased separately from the SS7 subsystem software (cktint and ISUP). The TCAP software consists of one floppy disk for SEPT. To install TCAP, complete the following steps:

1. If you haven’t already done so, log in to the SS7 subsystem as **root**.
2. **cd** to **/export/home**.

3. Enter the following command and press **Return**:

   /etc/init.d/volmgt stop

4. Copy the SS7 subsystem TCAP SEPT floppy to the system drive:

   a. Insert the SEPT floppy in the SS7 subsystem drive.

   b. Enter the following command and press **Return:**

      cpio -icduv -C65536 -I/dev/rfd0

5. Enter the following command and press **Return**:

      /etc/init.d/volmgt start

6. When the system has copied the software, change the permissions on the SEPT install script by entering the following command and pressing **Return**:

      chmod 755 ./install_sept.sh

7. Enter the following command and press **Return**:

      ./install_sept.sh

   The following messages appear:

   Installing sept (File: ‘basename sept.cpio.Z)

   Creating sept password entry...

   Creating sept shadow entry...

   done

   Change password now...

8. Enter a password and press **Return**.

   When the SEPT software is completely installed, the following messages appear:

   Extracting files...

   done

   Setting up...

   done

9. Remove the SEPT diskette.

3.3 INSTALLATION CHECKLIST

This subsection summarizes the tasks that must be completed for an SS7 subsystem software installation. Items are listed in the order they should be completed.

> *NOTE: This list is independent of installing the VCO/4K generic software. Cisco Systems recommends that the VCO/4K generic software be completely installed prior to an SS7 software installation.*

Software Installation Task List

1. If you purchased TCAP software, follow the procedure in *Section 3.2* for software installation before proceeding. Your SS7 subsystem software was already loaded before system shipment.
2. Run **sys-config** for A side and follow the screen prompts (see *Section 3.4*).
3. If redundant, connect console to B-side SPARC and boot B-side SPARC.
4. Run **sys-config** for B side and follow the screen prompts (see *Section 3.4*).
5. Create/edit the **/etc/hosts** file for both sides (see *Section 3.5*).
6. Configure E1/T1 spans on VCO/4K (see *VCO/4K System Administrator's Guide*):
    - Clear channel
    - Class of Service = 2
    - Inpulse Rule = 0
    - Resource Groups
7. Test LAN and **/etc/hosts** via ping command from both sides if redundant system (see *Section 3.6*).
8. If you purchased the 2k-to-4k configurator and wish to run in 4K mode, install and run the configurator (see *Section 4.7*).
9. Configure for multiple SPs if desired (see *Section 4.3*).
10. Build MTP layer MML files on A side (see *Section 4.4*).
11. If running TCAP, build TCAP layer MML files on A side (see *Section 4.4*).
12. Build ISUP layer MML files on A side (see *Section 4.4*).
13. Create/build database area for NewNet using MML files on A side (see *Section 4.4*).
14. Build **$XNV/CktInt.cfg** file on A side (see *Section 4.5.1*).
15. Build **$XNV/ckt_ss7_to_sds** file on A side (see *Section 4.5.2*).
16. Build **$XNV/grp_ss7_to_sds** file on A side (see *Section 4.5.2.2*).
17. If running TCAP, build **$SNV/SeptCcDflt.cfg** file on A side (see *Section 4.5.3*).
18. Configure for redundancy (see *Section 4.6*).
19. Start NewNet stack, cktint, and SEPT (if running TCAP) on A side (see *Section 5.2*).

20. *If redundant*:
    - Copy all created files from above to B side and modify host names and VCO names as necessary (**skip Step 13**).

    NOTE: Cisco Systems recommends using FTP to copy files from one side to the other (utilize LAN connection). Use UNIX command "man ftp" for assistance.

    - *C*reate/build database area for NewNet using MML files on B side.
    - *S*tart NewNet stack, cktint, and TCAP (if running TCAP) on B side (see *Section 5.2*).
21. Get data links up and aligned (**see note below**).
22. Pass SS7 messages to/from network (**see note below**).
23. *If redundant*, test redundancy switchovers. (Make sure the SS7 data links can be aligned on one side, switch over and have these links align on the other side after the switch over.)

    NOTE: ***Cisco Systems will not be able to completely install/test the SS7 installation without an SS7 network connection.*** *Only a customer-provided SS7 INET simulator can replace a network connection.*

    The level of testing performed by Cisco Systems Tech Support during an SS7 installation is minimal; it includes basic message passing (such as some circuit block messages) only.

    If testing message passing to and from the network is critical, please contact the Cisco Systems Consulting group to assist with this work. This group has the technical expertise and required equipment to perform in-depth testing. In addition, the Consulting group can offer a package to help with turning up your application to the network and testing the MTP and ISUP layers of the network (via simulation of the SS7 network and, in monitor mode, between the Cisco Systems SS7 system and the SS7 network).

3.4 RUNNING SYS-CONFIG

To prepare your system for operation, complete the following steps:

NOTE: If your SS7 subsystem console is a VT100 terminal, press **PF2** *on the numeric keypad instead of the* **F2** *key.*

1. Power up the VCO/4K with the SS7 subsystem installed.

   NOTE: It is normal for the warning message, "Unable to determine keyboard type," to appear on the SS7 subsystem's ASCII console when the system is powered up.

   When the reboot completes, the Terminal Type screen, shown in Figure 3.1, appears:

```
------------------------------------------------------------------------


What type of terminal are you using?
1)  ANSI Standard CRT
2)  DEC VT52
3)  DEC VT100
4)  Heathkit 19
5)  Lear Siegler ADM31
6)  PC Console
7)  Sun Command Tool
8)  Sun Workstation
9)  Televideo 910
10) Televideo 925
11) Wyse Model 50
12) Xterminal Emulator (xterms)
13) Other



------------------------------------------------------------------------
   Type the number of your choice and press Return:
```

Figure 3.1: Terminal Type Screen

2. Enter the number for your terminal type and press **Return**. The Specify Host Name screen, shown in Figure 3.2, appears.

   NOTE: Cisco Systems recommends a VT100 terminal.

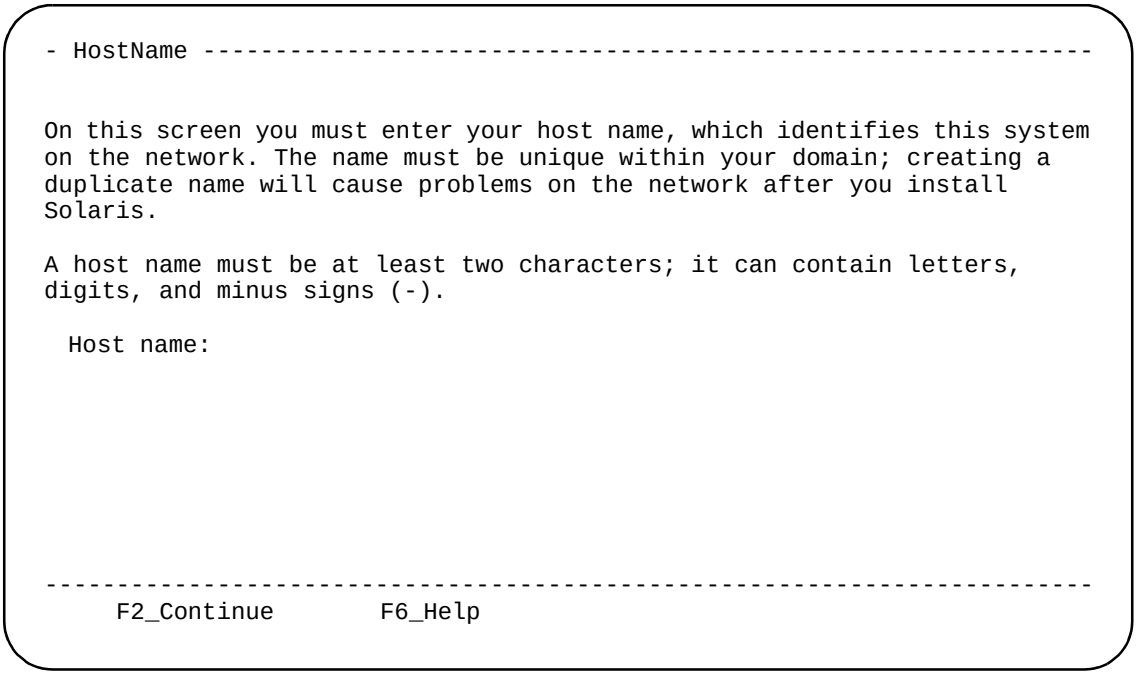


Figure 3.2: Host Name Screen

3. Enter the host name for the SS7 subsystem, following the instructions on the screen. Press **Return**. The following message appears:

   ```
   Press F2 to go to the next screen.
   ```

   Press **F2**. The Network Connectivity screen, shown in Figure 3.3, appears.

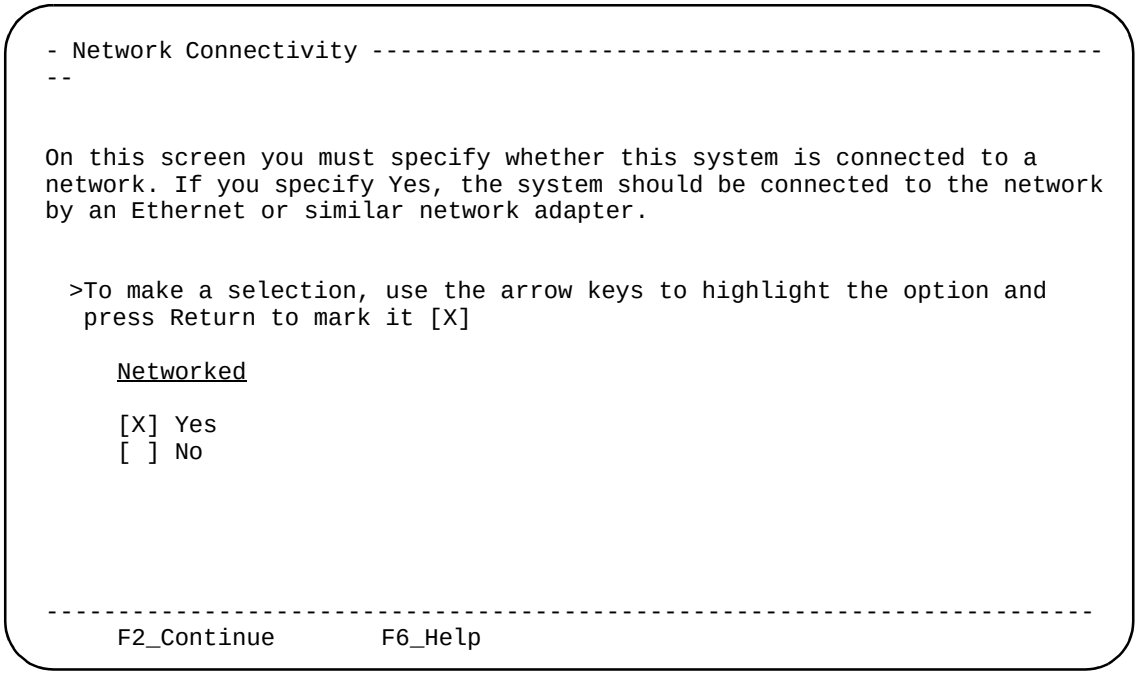


Figure 3.3: Network Connectivity Screen

4. Position the cursor in the correct field using the arrow keys and press **Return** to mark the field with an **X**.
5. Press Return. If you selected **Yes** in the Network Connectivity Screen, the IP Address screen, shown in Figure 3.4, appears.

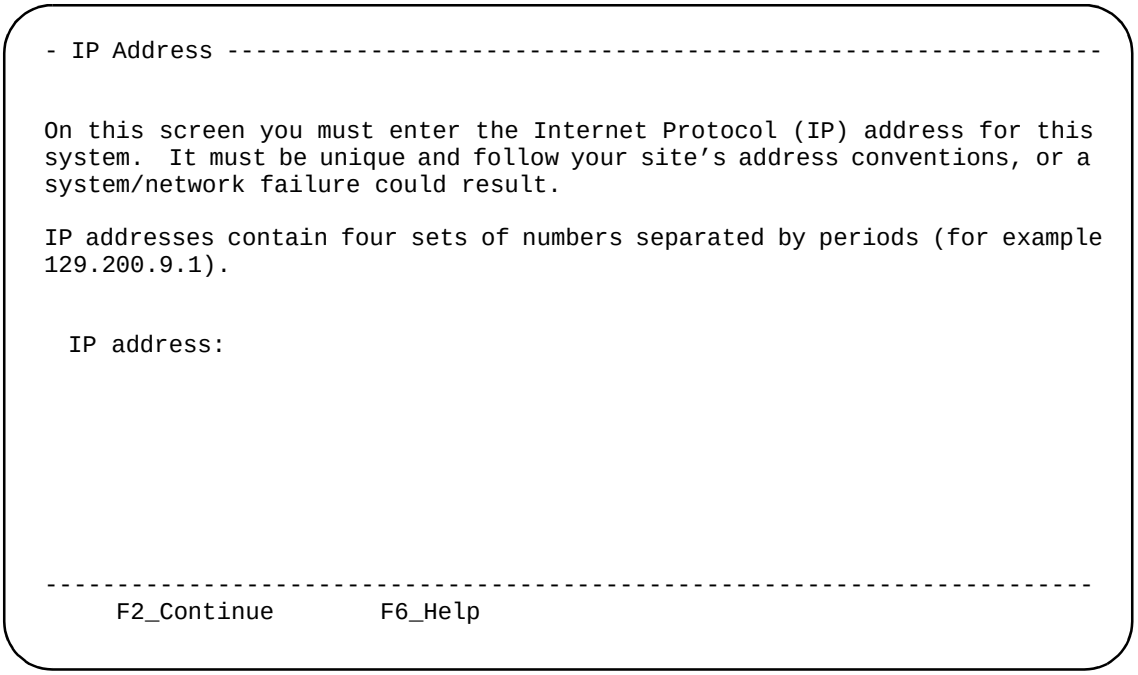


Figure 3.4: IP Address Screen

6. Enter the IP address for the SS7 subsystem, following the instructions on the screen, and press **Return**.

   The Confirm Host, Network, and IP Address screen, shown in Figure 3.5, appears.

```
- Confirm Information ------------------------------------------------


  Confirm the following information. If it is correct, press F2;
  to change any information, press F4.

  Host name:<your host≥
  Networked: Yes
  IP address: <host IP address>









-------------------------------------------------------------------------
     F2_Continue       F4 Change        F6_Help
```

Figure 3.5: Confirm Host, Network, And IP Address Screen

7. The Confirm Host, Network, and IP Address screen displays the hostname, network selection, and IP address you entered on the previous screens.

   To accept the displayed information and continue the configuration, press **F2**.

   The Name Service screen, shown in Figure 3.6 appears.

   To change the hostname, press **F4**. The Host Name screen (Figure 3.2) reappears. Go back to Step 3.

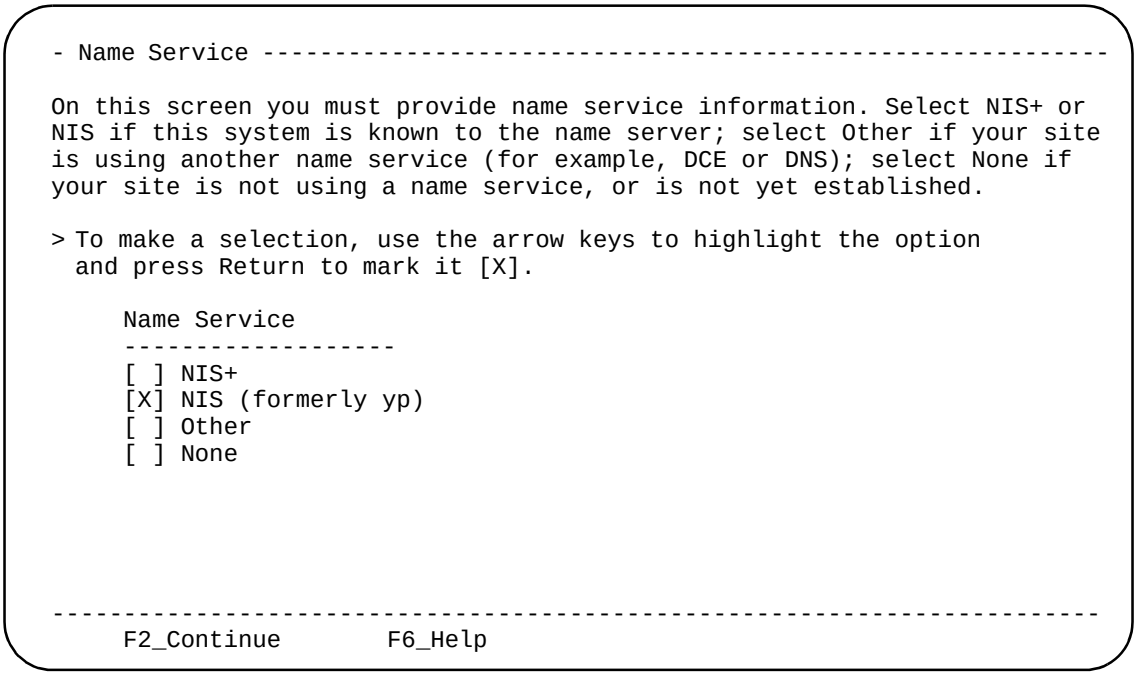


Figure 3.6: Name Service Screen

8. Specify the name service for your system:
   a. Use the up or down arrows to highlight the name service.
   b. Press **Return**. Your selection is marked with an **X**.
   c. Press **F2** to continue. The Domain Name screen, shown in Figure 3.7, appears if you enter anything except "None." Otherwise, continue to Step 13.

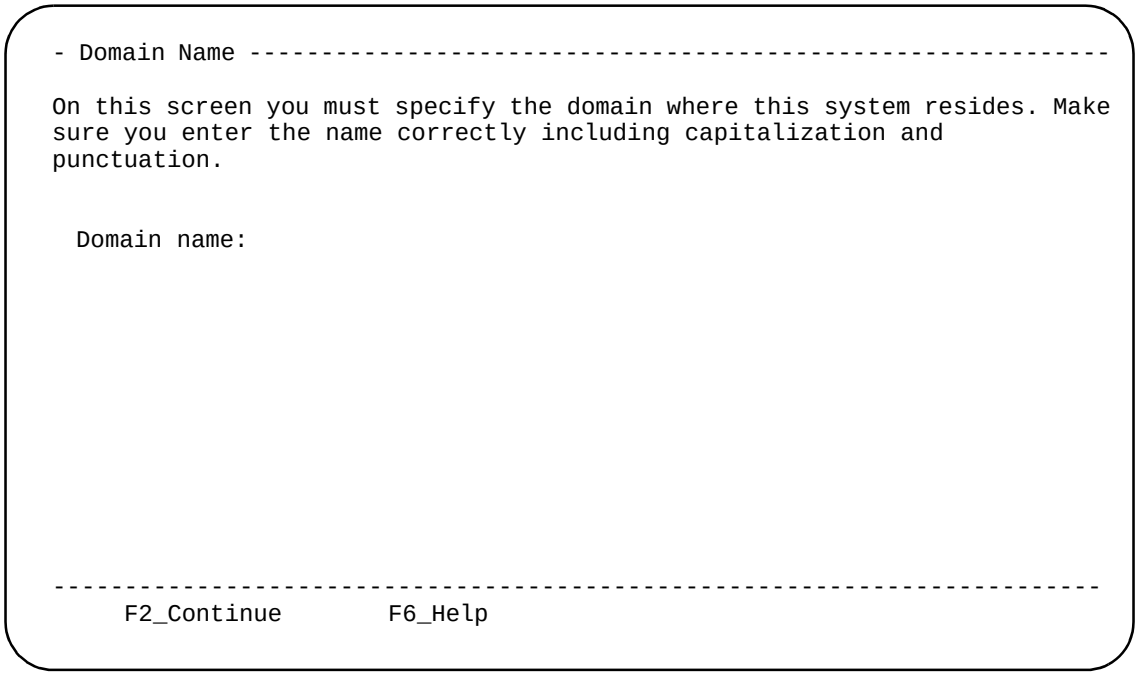


Figure 3.7: Domain Name Screen

9. Enter the domain name in which the SS7 subsystem resides following the instructions on the screen.

   Press **F2** to continue. The Name Server screen, shown in Figure 3.8, appears.

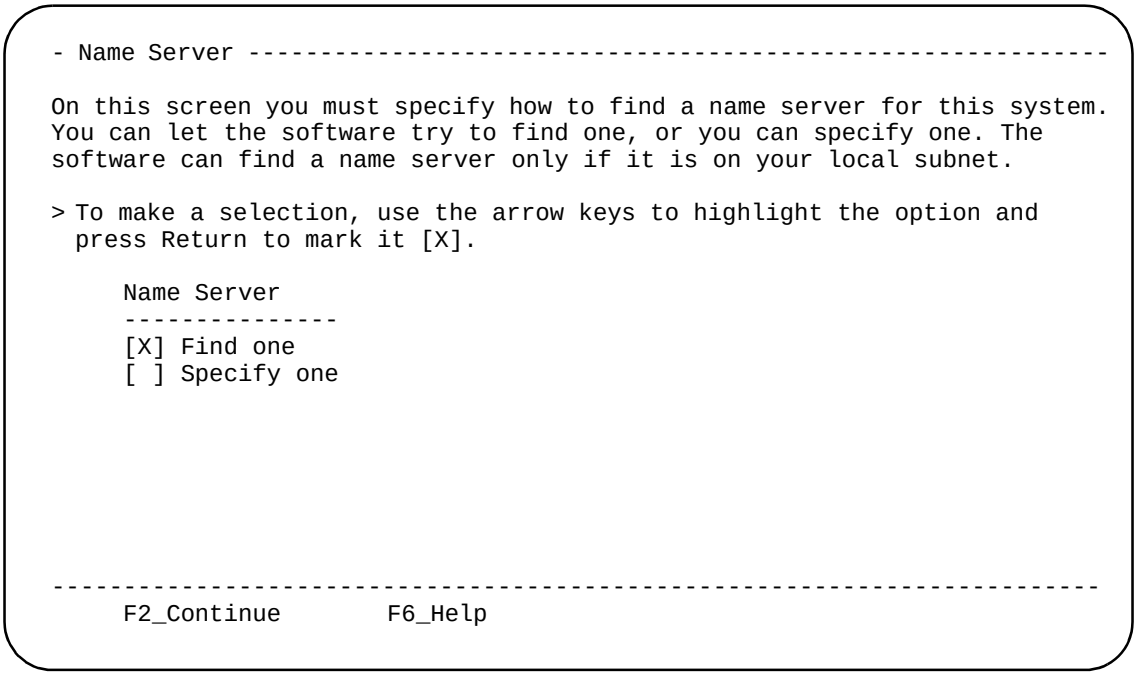


Figure 3.8: Name Server Screen

10. The Name Server screen allows you to specify a Name Server or allows the system to find one for you.

    To let the system find a name server for you:

    a. Press **F2** to continue. The Confirm Host Information screen, shown in Figure 3.10, appears.

    b. Go to Step 12.

    To specify a name server:

    c. Use the down arrow to highlight the **Specify one** field.

    d. Press **Return**. The field is marked with an **X**.

    e. Press **F2**. The Specify Name Server Information screen, shown in Figure 3.9, appears.

```
- Name Server Information ------------------------------------------------
------

On this screen you must enter the host name and IP address of your name
server. Host names must be at least two characters, and may contain
letters, digits, and minus signs (-). IP addresses must contain four sets
of numbers separated by periods (for example 129.200.9.1)

   Server's host name:
    Server's IP address:









------------------------------------------------------------------------
     F2_Continue       F6_Help
```

Figure 3.9: Specify Name Server Information Screen

11. Specify the Name Server's hostname and IP address and press **F2**. The Confirm Information screen, shown in Figure 3.10, appears.

```
- Confirm Information ----------------------------------------------------

> Confirm the following information. If it is correct, press F2;
  to change any information, press F4.


   Name service: NIS
   Domain name: <domain>
   Name server: Find one









------------------------------------------------------------------------
     F2_Continue       F4_Change       F6_Help
```

Figure 3.10: Confirm Information Screen

12. The Confirm Information screen displays the host information you entered on the three previous screens.

    To accept this information, press **F2**. The Subnet screen, shown in Figure 3.12 appears.

    To change any of the host information, press **F4**. The Name Service screen, (Figure 3.6) reappears. Go back to Step 8.

    NOTE: If the system determines that you made a mistake entering the host information, an error screen similar to the screen shown in Figure 3.11 appears.

```
- Name Service Error -------------------------------------------------

The following error occurred while trying to locate an NIS+ server for
domain <domain>:
The NIS+ server that responded is: .

To resolve this problem, run `nisinit -c -B' after the system has booted
or manually enter new name service information.

     Enter new name service information?
     ----------------------------------
     [X] Yes
     [ ] No









------------------------------------------------------------------------
     F2_Continue       F6_Help
```

Figure 3.11: Sample Name Service Error Screen

If you specify **Yes** *on this screen, the Name Service screen, (Figure 3.6) reappears. To correct this information, repeat Step 8 through Step 12.*

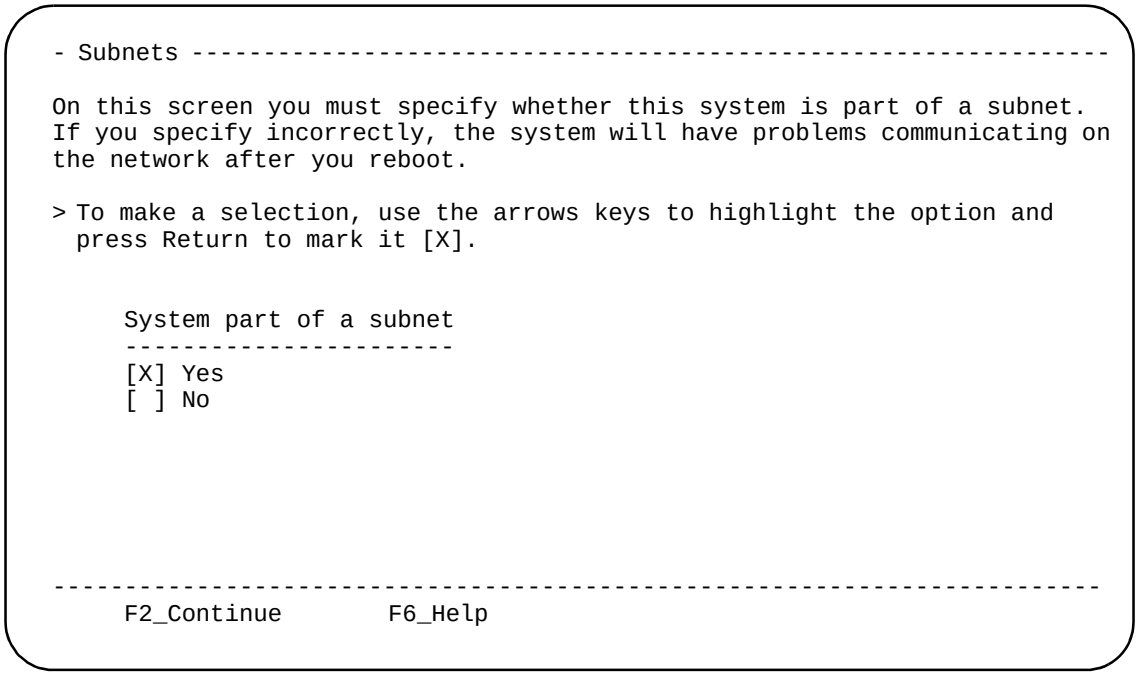


Figure 3.12: Subnet Screen

13. Specify whether your system is part of a subnet:
    a. Use the up or down arrows to highlight **Yes** or **No**.
    b. Press **Return**. Your selection is marked with an **X**.
    c. Press **F2**.

    If you selected **Yes**, the Netmask screen, shown in Figure 3.14, appears and you should go on to Step 15.

    If you selected **No**, a Confirm Information screen, as shown in Figure 3.13 appears.

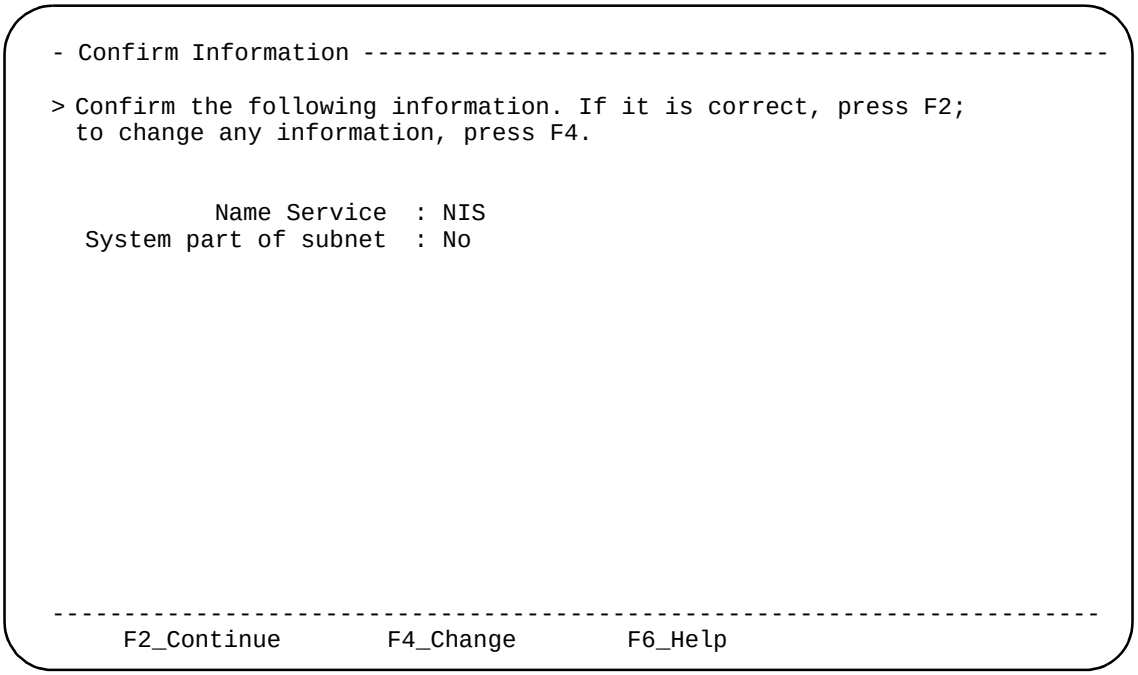


Figure 3.13: Confirm Information Screen

14. The Confirm Information screen displays the information you entered on the two previous screens.

    To accept this information, press **F2**. The Netmask screen, shown in Figure 3.14 appears.

    To change any of the information, press **F4**. The Subnet screen (See Figure 3.12) reappears. Go back to Step 13.

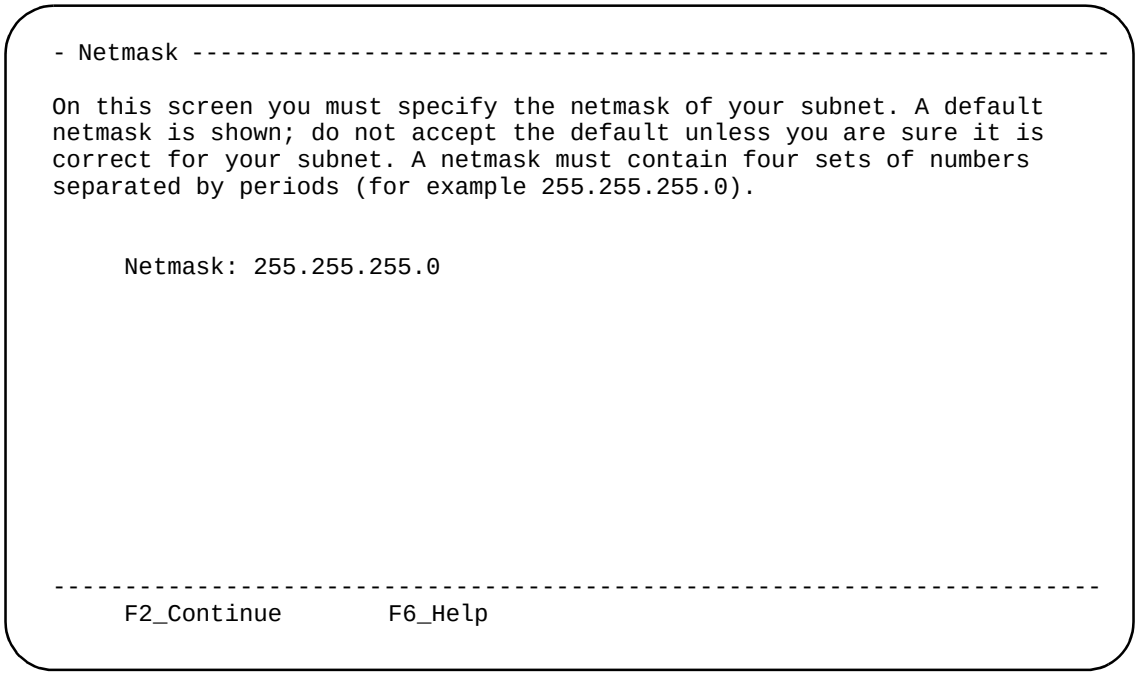


Figure 3.14: Netmask Screen

15. Enter the Netmask value and press **F2**. The Default Region screen, shown in Figure 3.15, appears.

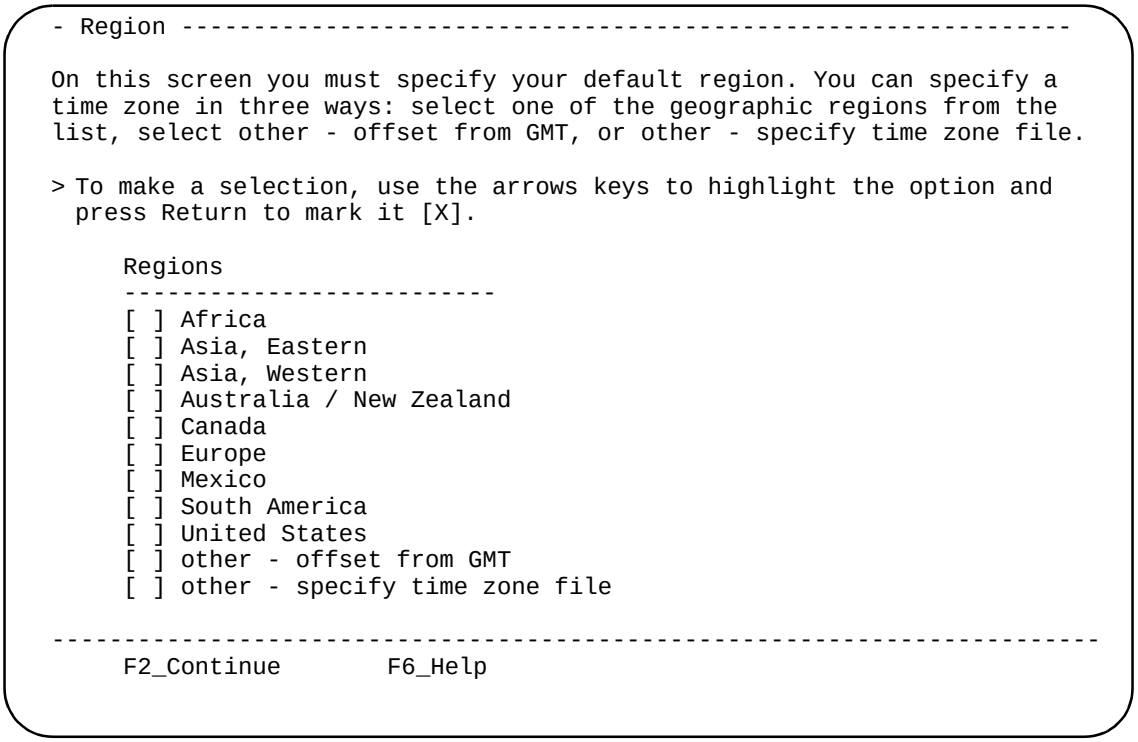


Figure 3.15: Default Region Screen

16. Specify the default region:
    a. Use the up or down arrows to highlight the region.
    b. Press **Return**. Your selection is marked with an **X**.
    c. Press **F2**. The Default Time Zone screen, similar to the screen shown in Figure 3.16, appears.

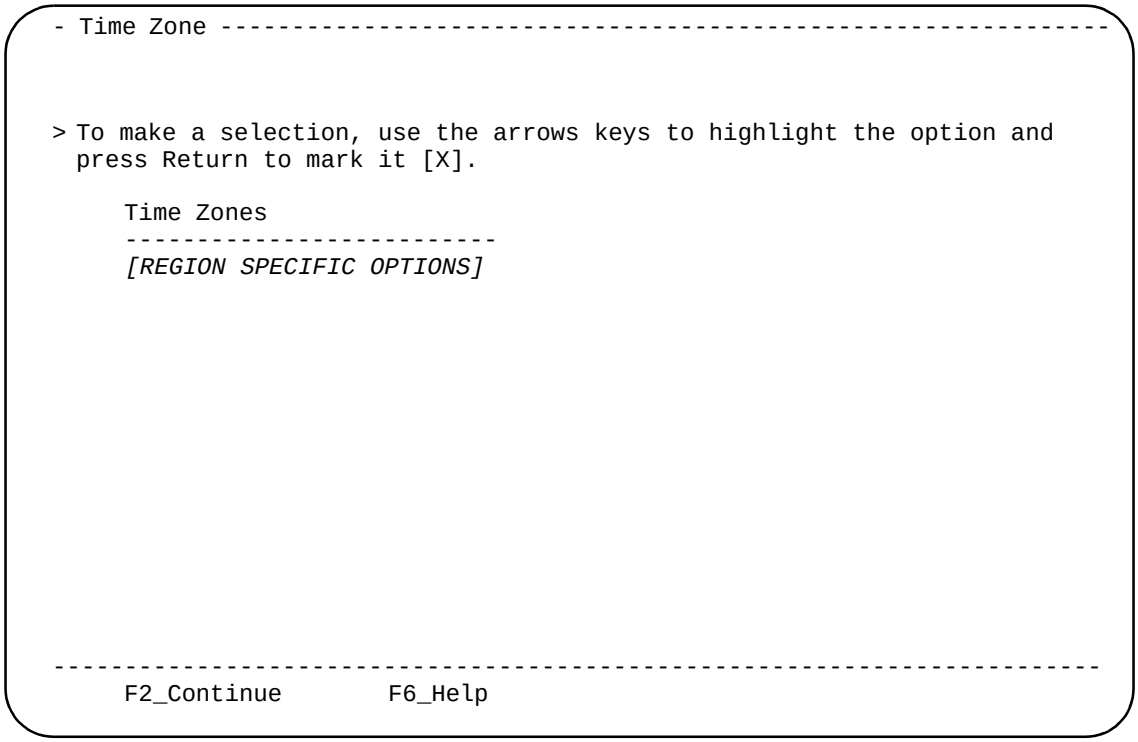


Figure 3.16: Default Time Zone Screen

17. Specify the default time zone:
    a. Use the up or down arrows to highlight the time zone.
    b. Press **Return**. Your selection is marked with an **X**.
    c. Press **F2**. The Default Date and Time screen, similar to the screen shown in Figure 3.17, appears.

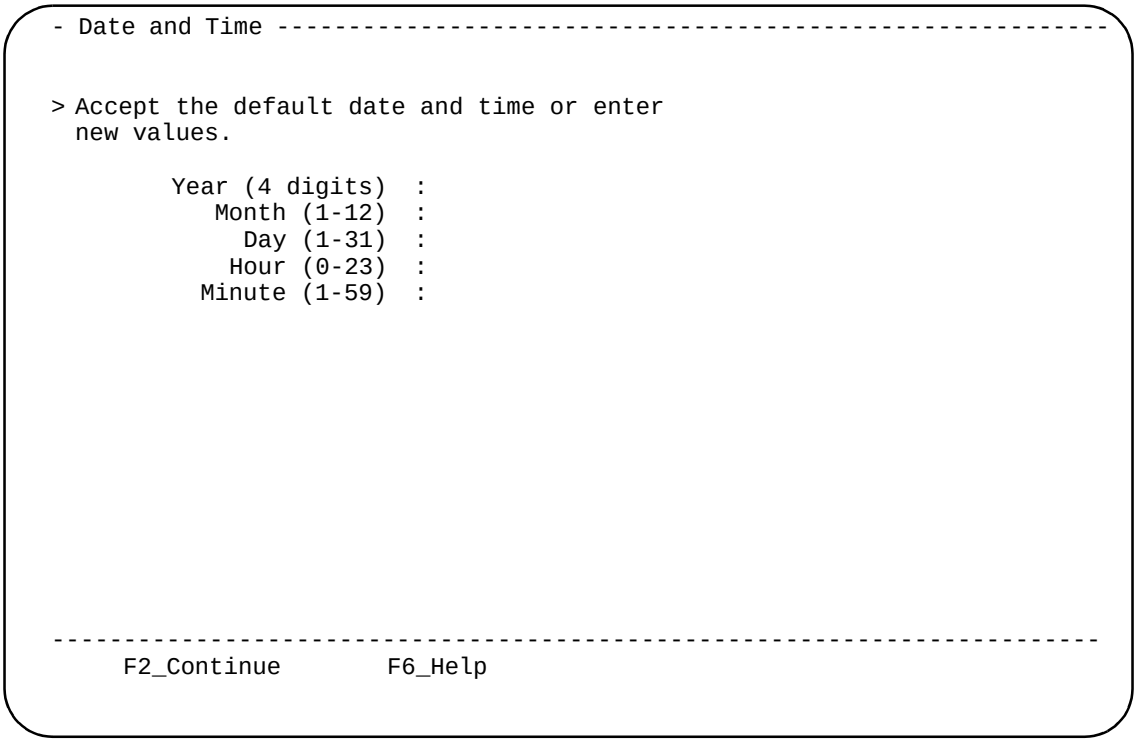


Figure 3.17: Default Date and Time Screen

18. Specify the default date and time:
    a. To accept the default date and time, press **F2**.
    b. To enter new values, use the up or down arrows to highlight the appropriate line and fill in new value.
    c. When finished, press **F2**. The Confirm Time Zone and Date and Time screen, similar to the screen shown in Figure 3.18, appears.

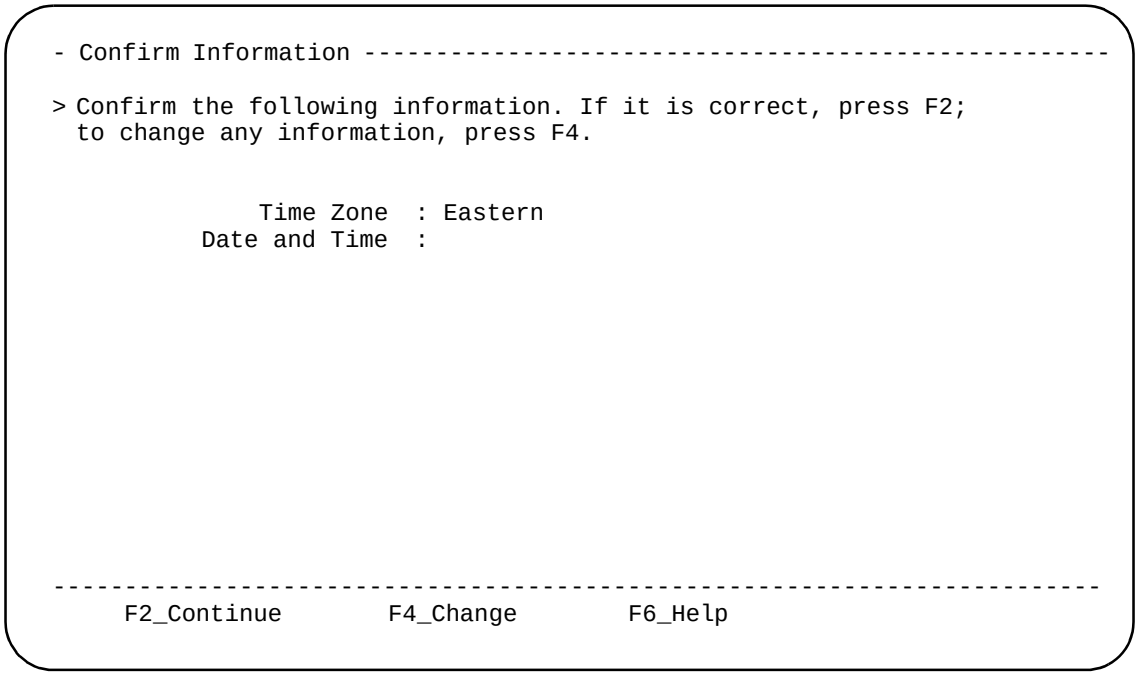


Figure 3.18: Confirm Time Zone and Date and Time Screen

19. The Confirm Time Zone and Date and Time screen displays the host information you entered on the three previous screens.

    To accept this information, press **F2**. The Specify Root Password screen, shown in Figure 3.19 appears.

    To change any of the information on the Confirm Time Zone and Date and Time screen, press **F4**. The Time Zone screen (See Figure 3.16) appears. Go back to Step 17.

```
On this screen you can create a root password.

A root password can contain any number of characters, but only the first eight
characters in the password are significant. (For example, if you create
'a1b2c3d4e5f6' as your root password, you can use `a1b2c3d4' to gain root
access.)

You will be prompted to type the root password twice; for security, the
password will not be displayed on the screen as you type it.

>  If you do not want a root password, press RETURN twice.

Root password:


Press RETURN to continue


Re-enter your password:


Press RETURN to continue.
```

Figure 3.19: Specify Root Password Screen

20. Enter the root password and press **Return**. The system prompts you to confirm your password by retyping it.

    NOTE: Cisco Systems recommends that you always enter a root password on this screen.

21. Reenter the password and press **Return**. If you make a mistake, the following message appears:

    ```
    Your password entries do not match. Try again.
    Press Return to continue.
    ```

22. When your root password is entered and confirmed, messages similar to the following appear:

```
System identification is completed.

Setting default interface for multicast: add net ###.#.#.#: gateway
<your host>
syslog service starting
Print Services started
volume management starting.
Cktint auto-start initialization in progress...This may take awhile.
Initialization n% completed.
done
The system is ready.
<host name> console login:
```

NOTE: If any of the sys-config information is incorrect, the boot messages after the "`System identification completed`*" message does not appear. If this occurs, press* **CTRL-D** *to reboot the system. When the system is ready, run* **sys-unconfig** *as described in Section 3.1.2 and reboot.*

23. Reboot the SS7 subsystem by typing **boot** and pressing **Return**.
24. If you have a redundant configuration, return to Step 1 and run the installation script from the console on side B.

The SS7 subsystem is now configured for two users: **root** and **cktint**.

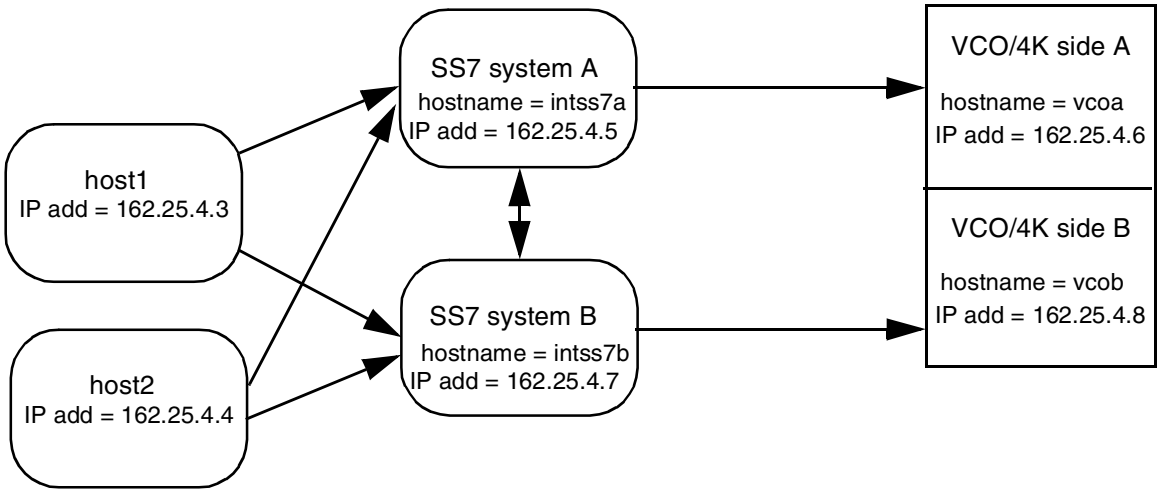
3.5 MODIFYING THE SS7 SUBSYSTEM /ETC/HOSTS FILE

In order for the SS7 subsystem to run on the network, it must have a hostname and an IP address defined in its **/etc/hosts** file. This information is used by the network to locate and identify the SS7 subsystem.

The **/etc/hosts** file is located in the system's root file system. When you ran **sys_config**, the script wrote the SS7 subsystem hostname and IP address into the **/etc/hosts** file for you. *You must modify the file to include the hostname and IP address of the VCO/4K to this file.*

If you have a redundant configuration, you must modify the file to include the SS7 subsystem hostname and IP address ***and*** *the hostname and IP address of the VCO/4K on the other side.* Figure 3.20 is an illustrated example of the **/etc/hosts** file for a redundant configuration with two host systems.

Block Diagram Of Redundant Configuration



Contents of /etc/hosts files

SS7 subsystem side A

127.0.0.1	localhost *
162.24.4.7	intss7b
162.24.4.5	intss7a
162.25.4.8	vcoB**
162.25.4.6	vcoA
162.25.4.3	host1
162.25.4.4	host2

SS7 subsystem side B

127.0.0.1	localhost *
162.24.4.5	intss7a
162.24.4.7	intss7b
162.25.4.6	vcoA**
162.25.4.8	vcoB
162.25.4.3	host1
162.25.4.4	host2

* The localhost entry appears in all /etc/hosts files and is for use by the network for local loop-back functions. SS7 subsystem users can ignore this entry.

** Even though the SS7 system never communicates with the VCO/4K on its redundant side, Cisco Systems recommends that you include the hostname and IP addresses for Telnet purposes.

Figure 3.20: /etc/hosts Files in a Redundant Configuration

NOTE: In a redundant configuration, the host name for both SS7 subsystems must be the same except for the last character. The last character for side A must be lowercase "a." The last character for side B must be lowercase "b."

3.5.1 Modifying the /etc/hosts File for Non-Redundant Configurations

To modify the **/etc/hosts** file for a non-redundant system, complete the following steps:

1. From the SS7 subsystem console, log in as **root**.

   *NOTE: The initial password is factory-configured as "***abc123***" for* **root** *and* **cktint**. *However, the password may have been changed when the Circuit Interworking software was installed. If you do not know* **root***'s password, contact your network administrator.*

2. Enter **cd** to **/etc**.
3. Open the **hosts** file for editing with a UNIX text editor, such as **vi**.
4. Add the hostnames and the IP addresses for the VCO/4K and host(s). Refer to your Network Information Worksheet for this information.
5. Save and close the file.

   NOTE: You do not have to reboot the system for these changes to take affect.

3.5.2 Modifying the /etc/hosts File for Redundant Configurations

To modify the **/etc/hosts** file for a redundant system, complete the following steps:

1. From the SS7 subsystem console on side A, log in as **root**.

   *NOTE: The initial password is factory-configured as "***abc123***" for* **root** *and* **cktint**. *However, the password may have been changed when the Circuit Interworking software was installed. If you do not know* **root***'s password, contact your network administrator.*

2. Enter **cd** to **/etc**.
3. Open the **hosts** file for editing with a UNIX text editor, such as **vi**.
4. Add the hostnames and the IP addresses for the side A VCO/4K, host(s), and the SS7 subsystem installed on side B (See Figure 3.20).
5. Save and close the file.
6. From the SS7 subsystem console on side B, log in as **root** and repeat Step 2 and Step 3.
7. Add the hostnames and the IP addresses for the side B VCO/4K, host(s), and the SS7 subsystem installed on side A (See Figure 3.20).
8. Save and close the file.

   NOTE: You do not have to reboot the system for these changes to take affect.

3.6 TESTING THE INSTALLATION

This section describes how to test the installation by using the UNIX **ping** command. Refer to your Network Information Worksheet for the correct hostnames for these procedures.

3.6.1 Testing Non-Redundant Configurations

To verify that the network connections and **/etc/hosts** entries are correct, complete the following steps:

1. Verify that all systems are up and running and that **root** is still logged into the SS7 subsystem.
2. From the SS7 subsystem console, use the **ping** command as follows:

   ping **<hostname>**

   where <hostname> is the name you assigned to the VCO/4K.

   If the connections and /etc/hosts entries are correct, the system responds to the ping command with the following message:

   <hostname> is alive

   NOTE: If there is a problem, the system responds with one of the following error messages:

   no answer from **<hostname>**

   or

   unknown host **<hostname>**

   *If either of these messages appear, verify that the hostnames and the IP addresses in the **/etc/hosts** file are correct. If these are correct, verify the network connections. If you have any questions, contact Cisco Systems Technical Support.*

3.6.2 Testing Redundant Configurations

1. Go to the SS7 subsystem console on side A and, if necessary, log in as **root**.
2. Ping the side A VCO/4K using the following command:

   ping **<hostname>**

   where <hostname> is the name you assigned to the side A VCO/4K.

   If the connections and /etc/hosts entries are correct, the system responds with the following message:

   <hostname> is alive

3. Ping side B SS7 subsystem installed in side B.
4. Go to the SS7 subsystem console on side B.
5. Ping the side B VCO/4K.
6. Ping the side A SS7 subsystem.

   NOTE: If there is a problem, the system responds with one of the following error messages:

   no answer from **<hostname>**

   or

   unknown host **<hostname>**

   *If either of these messages appear, verify that the hostnames and the IP addresses in the **/etc/hosts** file are correct. If these are correct, verify the network connections. If you have any questions, contact Cisco Systems Technical Support.*

Section 4
SS7 SUBSYSTEM CONFIGURATION

4.1 INTRODUCTION

This section describes how to configure the following SS7 subsystem components:

- Multi-SP Configuration
- SS7 layers (Message Transfer Part (MTP), ISUP, and TCAP)
- Circuit Interworking (CktInt)
- Resource provisioning files
- Redundancy
- 2k to 4k and 4k to 2k Configurator

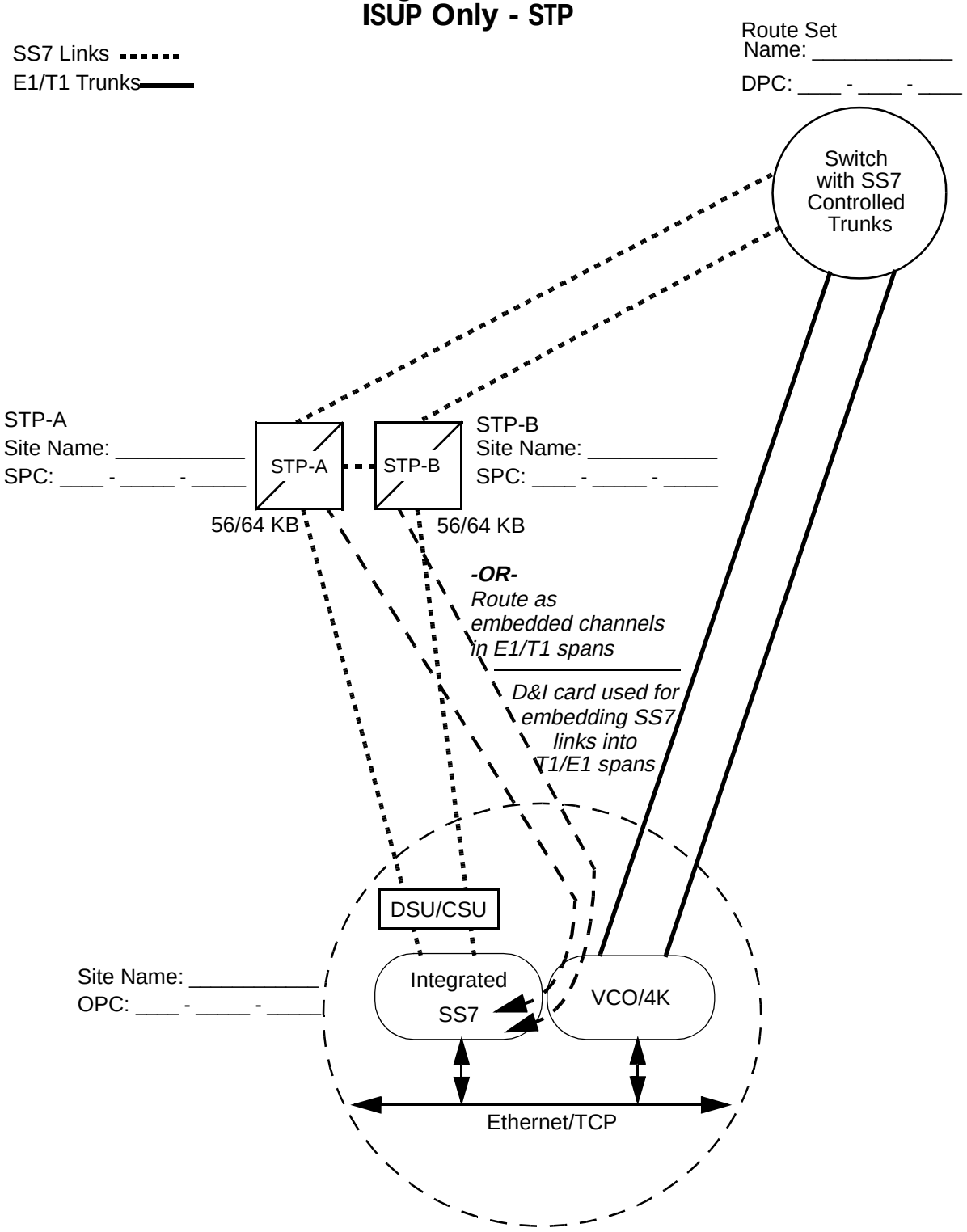
The SS7 subsystem software includes several default configuration files which you can modify to suite the needs of your installation. The SS7 layer configuration files are .mml files. The CktInt and resource provisioning files are text files, which you modify with any standard UNIX text editor. You can also display and modify these files with the Configuration Tool.

4.2 CONFIGURATION WORKSHEETS

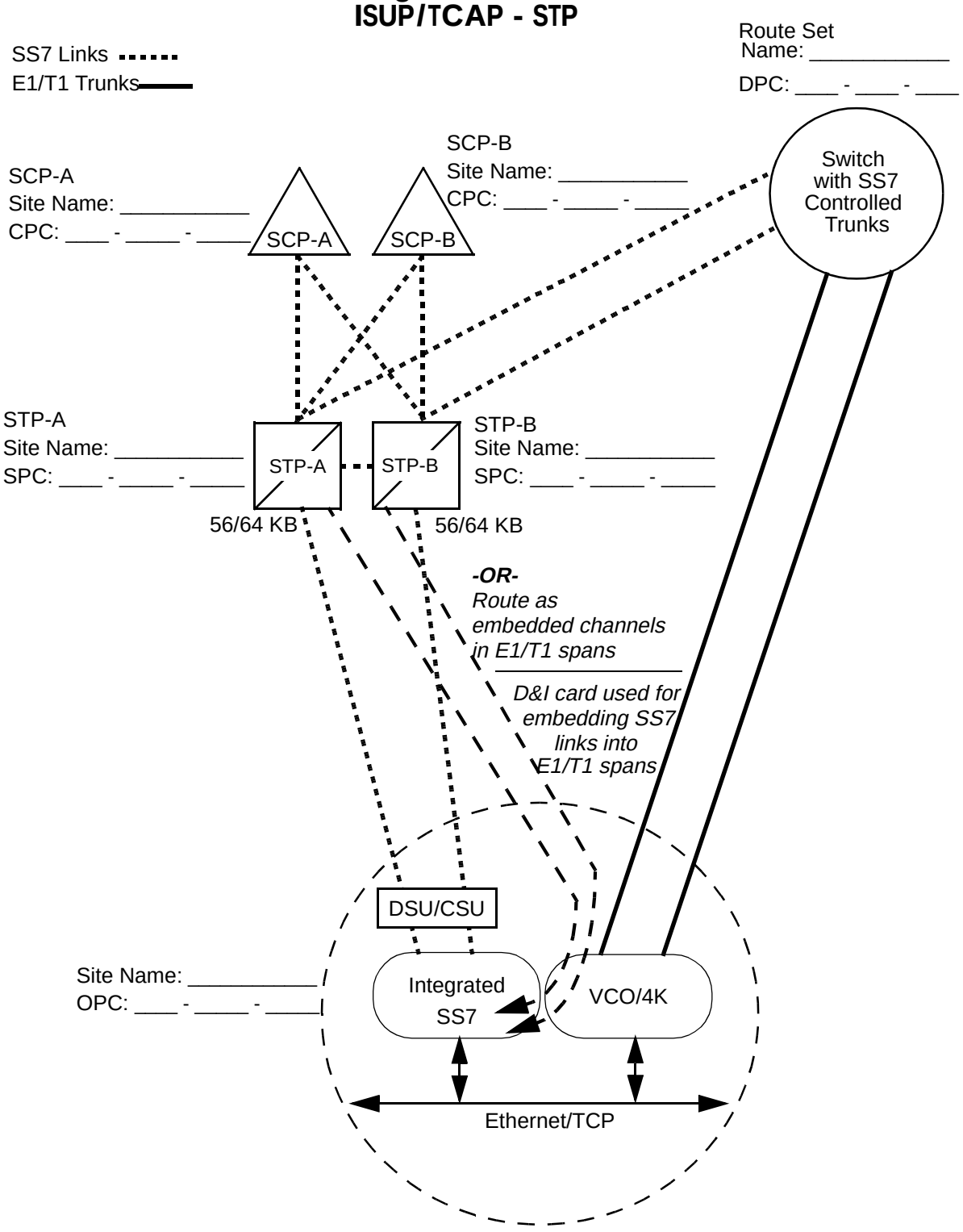
Before you begin the configuration, make copies of the appropriate Site Configuration and Circuit Configuration worksheets, which follow, and fill them out with the assistance of your network administrator. You also need the Network Information worksheets from *Section 3*.

If you have any questions regarding the configuration worksheets, contact Cisco Systems Technical Support for assistance.

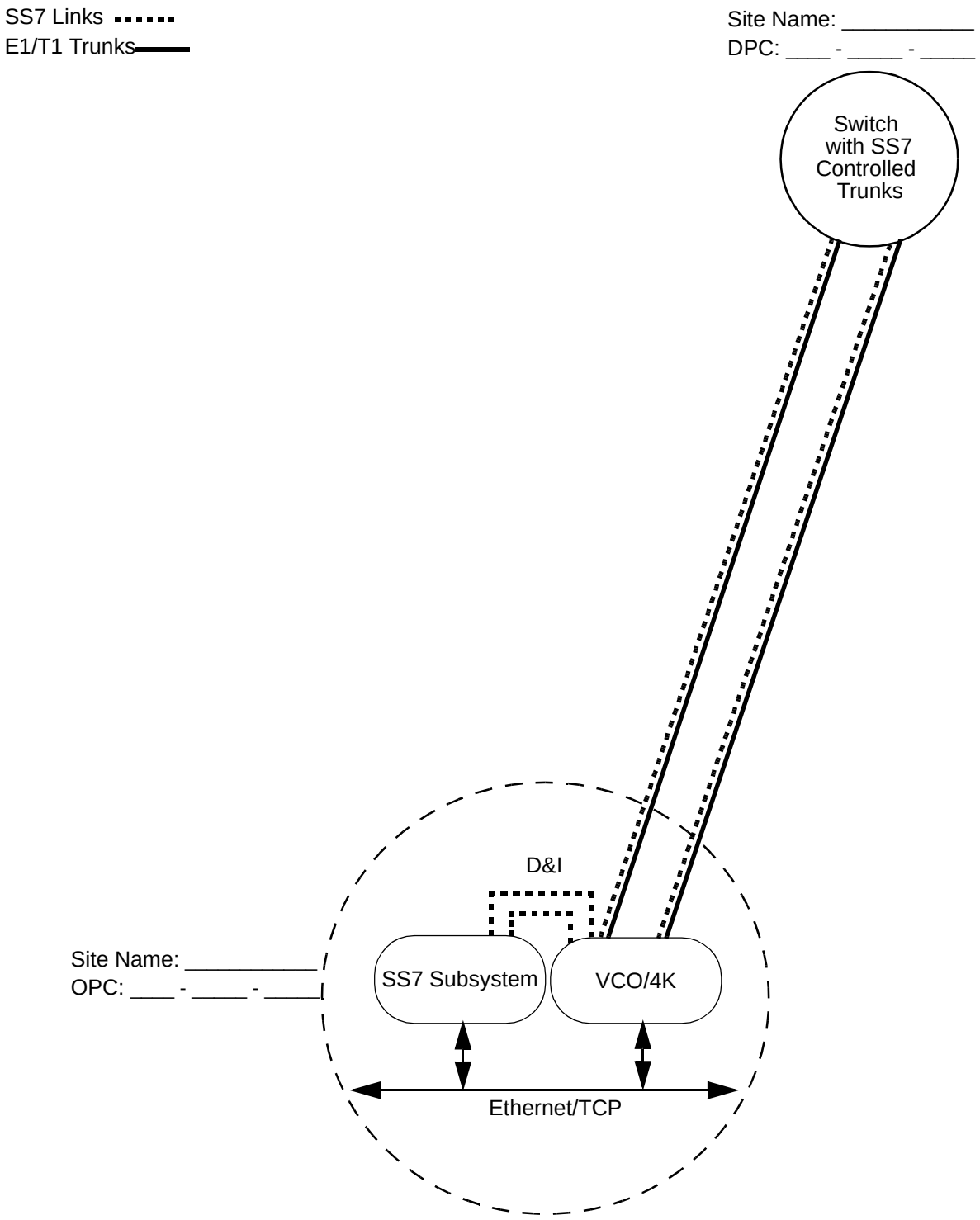
Site Configuration Worksheet
ISUP Only - STP



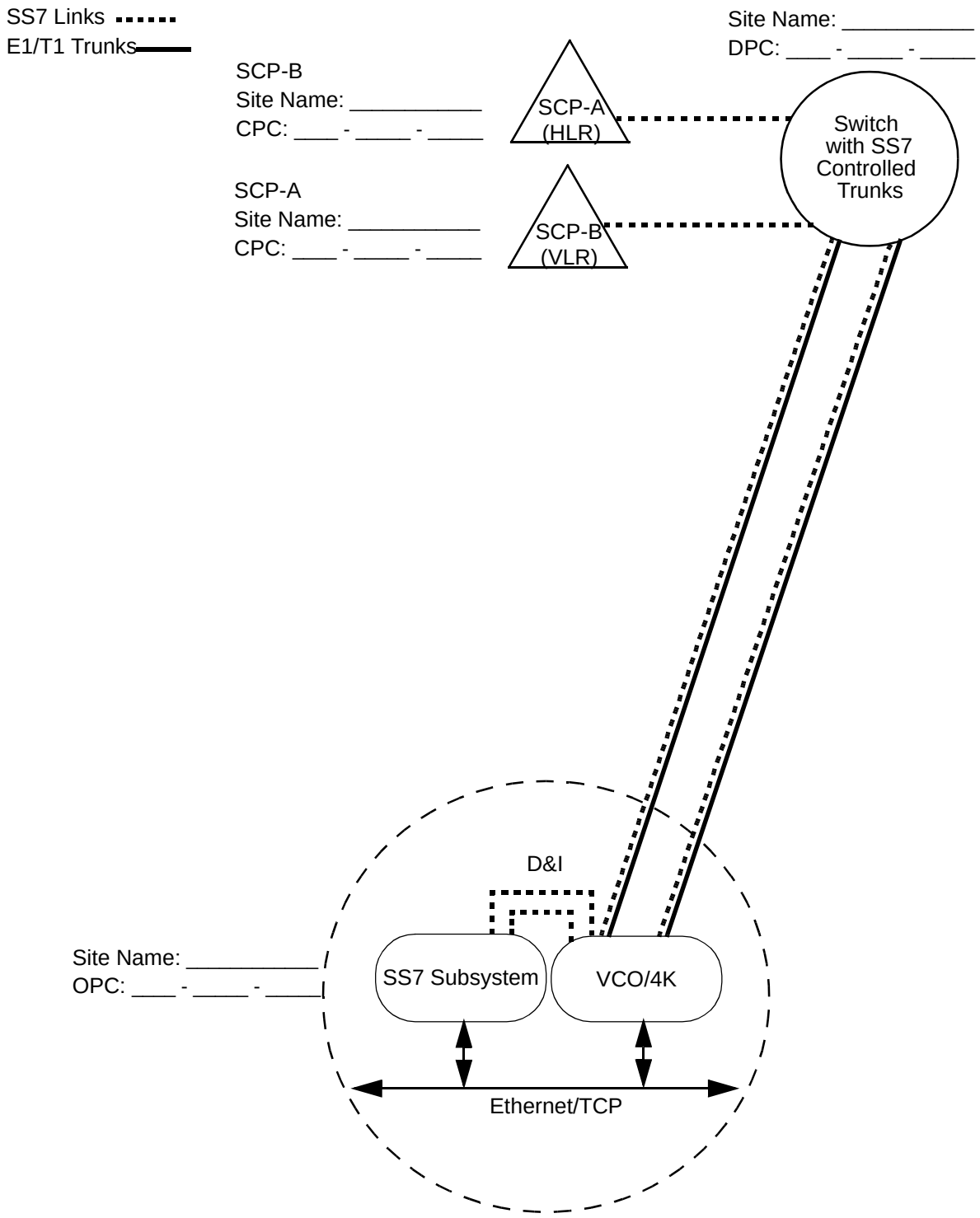
Site Configuration Worksheet
ISUP/TCAP - STP



Site Configuration Worksheet
ISUP Only - No STP



Site Configuration Worksheet
ISUP/TCAP - No STP



Circuit Configuration Worksheet

VCO/4K Port Addr.	Circuit ID (CCTNUM)	Circuit Name	Circuit Group (GRPID)	Trunk ID (TRNKGRPID)	Resource Group	Rack	Level	Slot

4.3 MULTI-SP CONFIGURATION

You may have up to eight different SPs, however, you cannot support link-level redundancy if you exceed four SPs (the SS7 system supports a maximum of eight links). Valid SP values are 0 to 7; you *must start with 0* and increment by ones (i.e. 0, 1, 2...).

Each SP may specify a different variant (i.e., SP 0 = Generic, SP 1 = Spain, and SP 2 = Australia), but it is not possible to have variants from different SS7 stacks (ANSI and ITU variants cannot exist together).

NOTE: If you want to use the multi-SP feature, you must first purchase and install additional EBS SS7 software licenses and create new configuration files. Then, follow the directions below.

4.3.1 Single-SP to Multi-SP Configuration

1. Log in as **cktint** to the SS7 subsystem side A, **<hostname>a**.
2. Enter the following command and press **Return**:

   configure-msp.sh

   The following message appears:

   Do you want to configure for MultiSP? (y/n):

3. To choose multiple-SP configuration, type **y** and press **Return**. The following message appears:

   Please Enter Signalling Point Numbers To Start Separated by Space:

4. Type the SP numbers you want to run and press **Return**.

   0 1 2 3

   The following message appears:

   The System is Configured for MultiSP. Please reboot the system

5. Log in as **root**.
6. Reboot your SS7 subsystem side A.
7. If you have a redundant system, log in as **cktint** to the SS7 subsystem side B, **<hostname>b**, and repeat Step 2 through Step 5.
8. Reboot your SS7 subsystem side B.

   NOTE: SP numbers must be the same on both sides.

4.3.2 Multiple-SP to Single-SP Configuration

1. Log in as **cktint** to the SS7 subsystem side A, **<hostname>a**.
2. Enter the following command and press **Return**:

   configure-msp.sh

   The following message appears:

   Do you want to configure for MultiSP? (y/n):

3. To choose single-SP configuration, type **n** and press **Return**. The following message appears:

   The System is configured back to Single SP.Please reboot the system

4. Log in as **root**.
5. Reboot your SS7 subsystem side A.
6. If you have a redundant system, log in as **cktint** to the SS7 subsystem side B, **<hostname>b**, and repeat Step 2 through Step 4.
7. Reboot your SS7 subsystem side B.

4.4 SS7 LAYERS CONFIGURATION

The configuration files for the SS7 layers are .mml files. They are Man-Machine Language (MML) commands described in the *NewNet Access Manual*. This subsection contains an example configuration file and instructions for creating and loading the configuration file for your site. This subsection is divided into three parts:

- *Section 4.4.1* describes how to use the MML HELP command.
- *Section 4.4.2* contains examples of the configuration files and describes the MML commands used to create them. (For a complete listing of all MML commands and more examples, refer to the *NewNet Access Manual*.)
- *Section 4.4.3* describes how to create/modify and load the configuration file on the SS7 subsystem.

Example configuration files are located in **~/sys/SPcc ($SPC)**. Use the examples in *Section 4.4.2* and the example files in **~/sys/SPcc** as a guide for creating the configuration file for your site.

NOTE: Lines in the configuration files must be ordered as shown in these examples.

Place your custom .mml files in **~/sys/SPcc ($SPC)** for signaling point (SP) 0 or single-SP configurations.

If you want to use more than one SP for applications such as gateways, *you must create a new file and place it in the appropriate directory (see Section 4.4.3) for each additional SP* (i.e., second SP file in **~/sys/SPcc1 ($SPC1)**, third SP file in **~/sys/SPcc2 ($SPC2)**, etc.).

4.4.1 MML HELP Command

The HELP command allows you to view on-line documentation for any command while you are running MML. To view the on-line documentation, complete the following steps:

1. If you are not running MML, enter the following command and press **Return**:

   ```
   mml 0
   ```

   The **MML>** prompt appears.

2. At the **MML>** prompt, enter the following command and press **Return**:

   ```
   HELP:;
   ```

   A listing of all the MML commands appears. At the end of the listing, a new prompt, **MML_HELP>**, also appears.

3. At the **MML_HELP>** prompt, enter the command, exactly as it appears in the listing, followed by a colon (:) and semicolon (;), and press **Return**.

   The on-line documentation for the command appears.

   Once you have reached the end of the on-line documentation file, the system automatically exits HELP and the **MML>** appears.

4. To view the on-line documentation for additional commands, repeat Step 2 and Step 3.

4.4.2 Site Configuration Example

There are three files for configuring the SS7 layers required to run ISUP and TCAP:

- Message Transfer Part (MTP) configuration file, which has two distinct parts:
  - — MTP Level 2 Provisioning, which configures the SS7 subsystem signaling points and link sets.
  - — MTP Level 3 Provisioning, which configures the route sets.
- TCAP Provisioning—defines the SCCP configuration used for TCAP message routing.
- ISUP Provisioning—configures the T1 (or E1) circuits.

The MTP and ISUP configuration files are used by the system for ISUP applications. The MTP and SCCP configuration files are used by the system for TCAP applications. You can use the same MTP files for both applications, or you can have separate MTP files for each application, depending on your network configuration.

Figure 4.1 is an illustrated example of a site configuration with ISUP and SCCP using the same links.

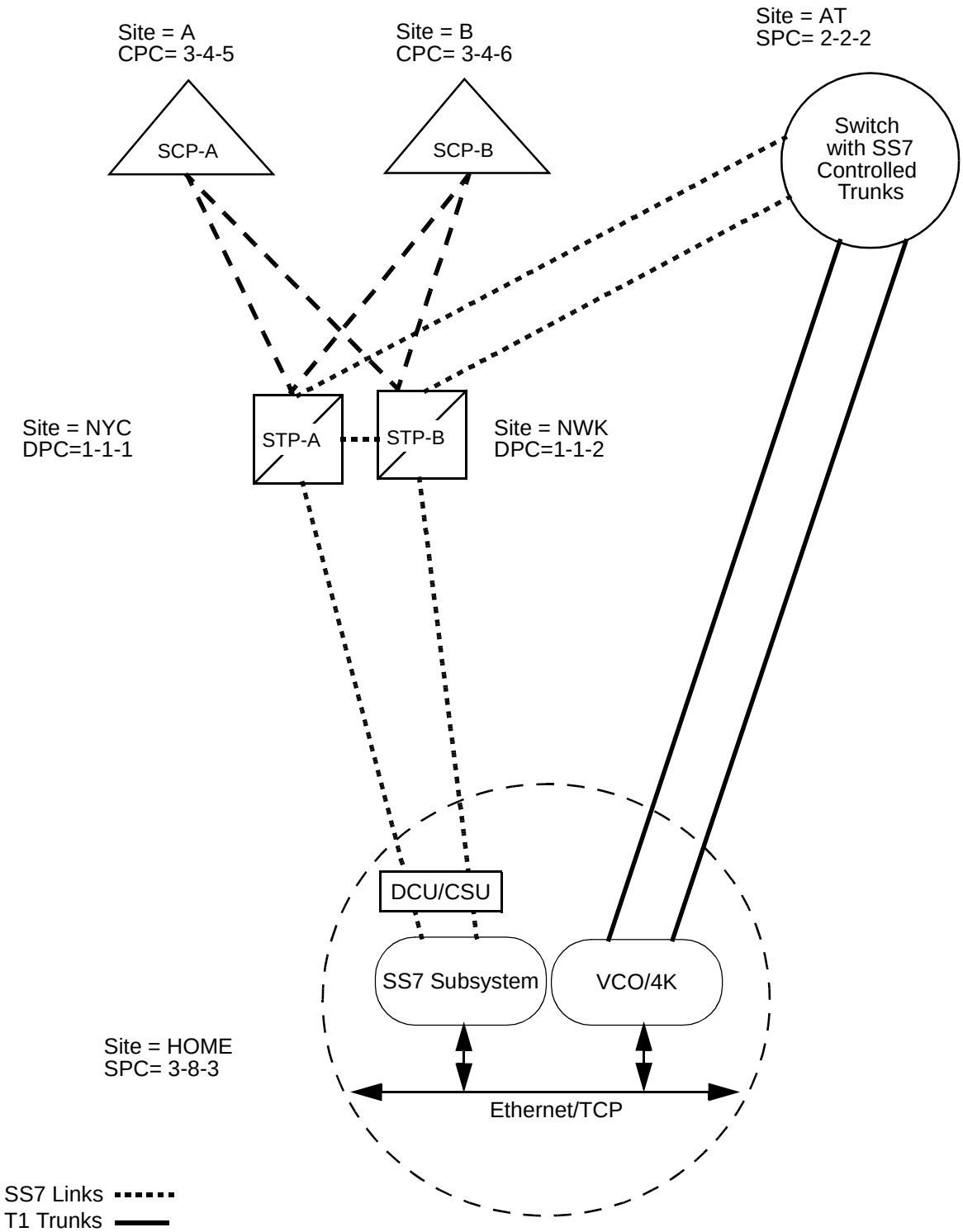
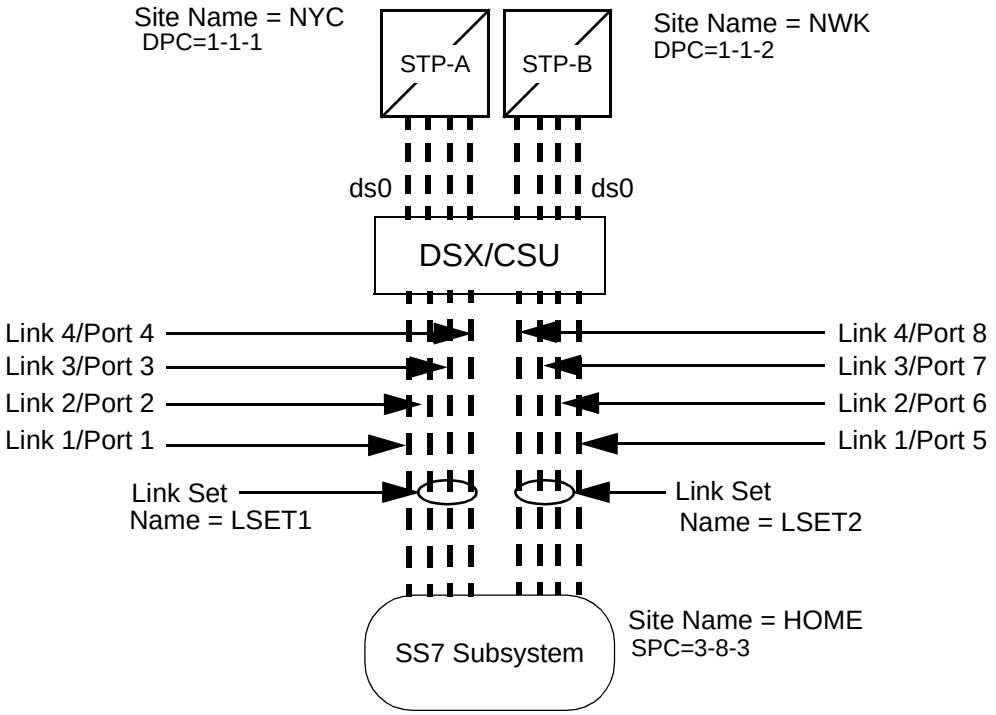


Figure 4.1: Site Configuration Example

4.4.2.1 MTP Level 2 Provisioning Part

The MTP Level 2 Provisioning part of the MTP configuration file defines the network indicator and signaling point code used by the SS7 subsystem, and defines the SS7 subsystem links and link sets. Figure 4.2 is an MTP Level 2 provisioning example.



Lines in **<filename>**, where filename is recommended to be called **mtp.mml** (or any name that includes mtp and has the extension .mml) for MTP Level 2 Provisioning.

```
(1) MML-CONFIG:LOG=ON,TIMEOUT=6000;
(2) MODIFY-SP:NAME=HOME,SPC=3-8-3,NI=NATIONAL,TYPE=SEP,RESTART=OFF,
    SLTC=REGULAR;
(3) ADD-LSET:LSET=LSET1,DPC=1-1-1,LOADED=4,ACTIVE=4,TYPE=ALINK,BR=64000;
(4) ADD-LSET:LSET=LSET2,DPC=1-1-2,LOADED=4,ACTIVE=4,TYPE=ALINK,BR=64000;
(5) ADD-LINK:LINK=LINK1,LSET=LSET1,SLC=0,PORT=1,TYPE=DTE,PRIORITY=0;
(6) ADD-LINK:LINK=LINK2,LSET=LSET1,SLC=1,PORT=2,TYPE=DTE,PRIORITY=1;
(7) ADD-LINK:LINK=LINK3,LSET=LSET1,SLC=2,PORT=3,TYPE=DTE,PRIORITY=2;
(8) ADD-LINK:LINK=LINK4,LSET=LSET1,SLC=3,PORT=4,TYPE=DTE,PRIORITY=3;
(9) ADD-LINK:LINK=LINK1,LSET=LSET2,SLC=0,PORT=5,TYPE=DTE,PRIORITY=0;
(10) ADD-LINK:LINK=LINK2,LSET=LSET2,SLC=1,PORT=6,TYPE=DTE,PRIORITY=1;
(11) ADD-LINK:LINK=LINK3,LSET=LSET2,SLC=2,PORT=7,TYPE=DTE,PRIORITY=2;
(12) ADD-LINK:LINK=LINK4,LSET=LSET2,SLC=3,PORT=8,TYPE=DTE,PRIORITY=3;
```

Figure 4.2: MTP Level 2 Provisioning Example

The **MML-CONFIG** command (line 1 of the example .mml file in Figure 4.2) turns on the command logging function. This function logs all MML commands into the file **$EBSHOME/access/RUN<SP#>/backup/MMLcmds.current**. The following are the parameters for MML-CONFIG:

LOG—Valid values are ON and OFF.

TIMEOUT—Specifies how long (in milliseconds) MML waits for an answer before timing out and displaying an SPM time out error message. Valid values for this parameter are 0 to 65000 milliseconds.

NOTE: Cisco Systems recommends you set the value to 6000 milliseconds.

The **MODIFY-SP** command (line 2) specifies the configuration for the SS7 subsystem and sets the SPC Restart State indicator and Signaling-Link Test-Control state indicator. The following are the parameters for MODIFY-SP:

NAME—The name of your site. Valid values are alphanumeric. Valid lengths are one to 11 characters.

SPC—The signaling point code (SPC) defined by the SS7 network administrator.

NI—The network indicator. Valid values for NI are:

INTERNATIONAL	00
SPARE (International Only)	01
NATIONAL	02
RESERVED (National Only)	03

TYPE—Specifies whether the SS7 subsystem is a signaling transfer point (STP) or signaling end point (SEP).

*NOTE: **The SS7 subsystem software only supports SEP**.*

RESTART—Sets the SPC Restart State indicator. Valid values are ON and OFF.

SLTC—Sets the Signaling-Link Test-Control State indicator. Valid values are REGULAR, SPECIAL, and OFF. Default is REGULAR.

NOTE: When modifying the signaling point code (line 2 of example), the RESTART parameter is automatically initialized as OFF. A second MODIFY-SP command must be entered to set the RESTART parameter to ON after initialization is complete.

The **ADD-LSET** command (lines 3 and 4) defines the link sets from the SS7 subsystem. A link set is a set of one or more links going to the same destination point code (DPC). The example in Figure 4.2 has two-link sets with four links in each set.

The following are the parameters for ADD-LSET:

LSET—A user defined name that identifies the link set. Valid values are alphanumeric. Valid lengths are 1 to 8 characters.

DPC—The DPC of the STP the link set to which the link set is connecting. Get the DPC from the SS7 network administrator.

LOADED—Defines how many links are configured in the link set (i.e., number of links from each STP). Valid values for SS7 subsystem configurations with four-link Sbus cards are 0 through 4. Valid values for SS7 subsystem configurations with eight-link Sbus cards are 0 through 8.

ACTIVE—Defines how many *active* links there are in the link set, that is, how many links will be carrying messages. Valid values for SS7 subsystem configurations with four-link Sbus cards are 0 through 4. Valid values for SS7 subsystem configurations with eight-link Sbus cards are 0 through 8.

NOTE: The NewNet Access Manual states that valid values for LOADED and ACTIVE are 0 to 15. However, SS7 subsystems configured with 4-link Sbus cards support a maximum of four links. Therefore, values greater than 4 are invalid. In the same way, SS7 subsystems configured with 8-link Sbus cards support a maximum of eight links. Therefore, values greater than 8 are invalid.

TYPE—Specifies the type of link set. The following are the valid values for TYPE:

Numeric Value	Alias	Meaning
0	ALINK	Access link
5	FLINK	Link between two SEPs

NOTE: The NewNet AccessManager manual states that 1 (BLINK), 2 (CLINK), 3 (DLINK), and 4 (ELINK) are also valid values for TYPE, but ***SS7 subsystem software only supports 0 (ALINK) and 5 (FLINK)****.*

BR—Specifies the baud rate for this link set. The following are the valid values for BR:

4800 bps

7200 bps

9600 bps

19200 bps

38400 bps

56000 bps

64000 bps

The **ADD-LINK** command (lines 5 through 12) defines the links that belong in each link set. The links in lines 5 to 8 belong to the link set you defined in line 3. The links in lines 9 to 12 belong to the link set you defined in line 4.

The following are the parameters for ADD-LINK:

LINK—A user-defined value that identifies the link. Valid values are alphanumeric. Valid lengths are 1 to 11 characters.

LSET—The name of the link set to which you are adding the link.

SLC—The signaling link code (SLC). SLC is a numerical value of 0 to 15. These values must be unique within each link set. You must start numbering at 0 and number each additional sequentially.

PORT—The logical port number for the ports on the Sbus card. The valid values for PORT are:

- 1 to 4 for 4-link configurations
- 1 to 4 and 5 to 8 for 8-link configurations

NOTE: The system uses ecpt file(s) to determine port numbers. Ports always begin at 1.

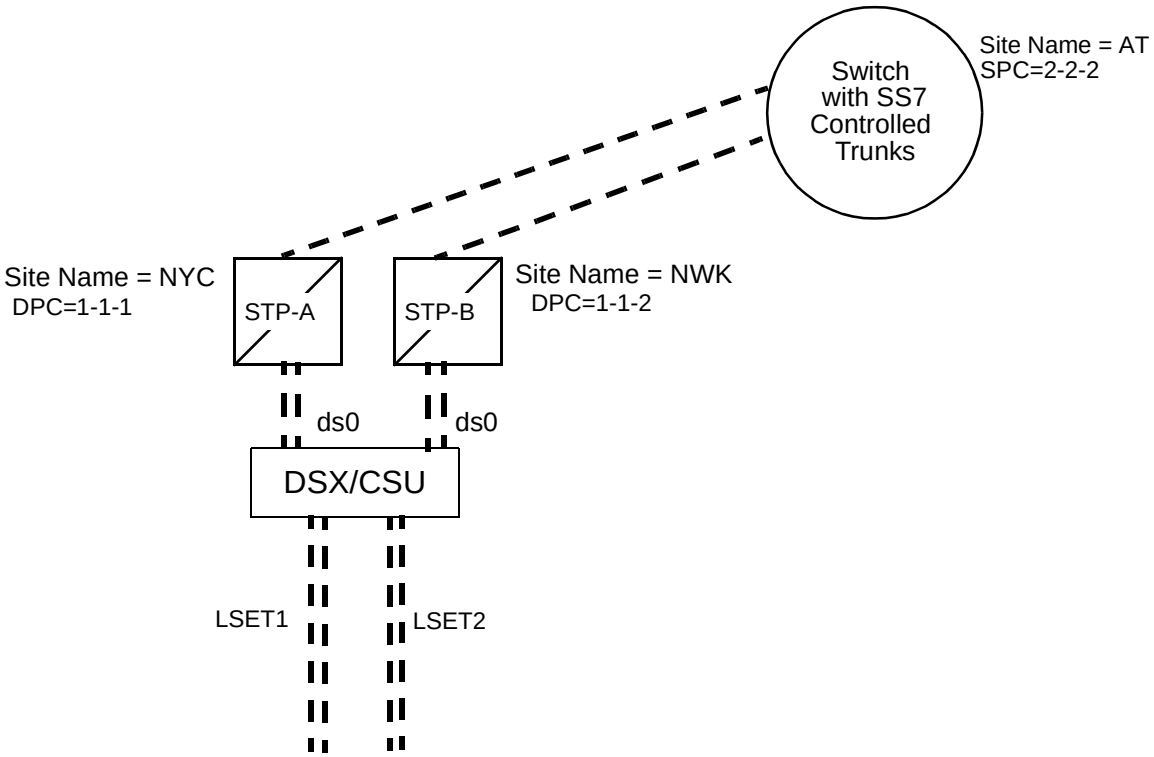
TYPE—This value is always DTE.

NOTE: The NewNet AccessManager manual states that TYPE can be either DTE or DCE, but ***SS7 subsystem software only supports DTE****.*

PRIORITY—The priority of the signaling link. Valid values are 0 to 15, where 0 is the highest priority and 15 is the lowest priority.

4.4.2.2 MTP Level 3 Provisioning

The MTP Level 3 Provisioning part of the MTP configuration file defines the routes. Figure 4.3 is an example of this part.



Lines in **<filename>** for MTP 3 Level Provisioning.

```
(13) ADD-RTSET:RTSET=AT,DPC=2-2-2,ROUTE1=LSET1,ROUTE2=LSET2;
(14) ADD-ROUTE:RTSET=LSET2STP,LSET=LSET1,PRIORITY=1;
(15) ADD-ROUTE:RTSET=LSET1STP,LSET=LSET2,PRIORITY=1;
(16) MODIFY-LSET:LSET=LSET1,ADMINSTATE=ACTIVE;
(17) MODIFY-LSET:LSET=LSET2,ADMINSTATE=ACTIVE;
(18) MODIFY-ALARM-CONFIG:DISPLAY=OFF;
(19) EXIT:;
```

Figure 4.3: MTP Level 3 Provisioning Example

The **ADD-RTSET** command in line 13 of Figure 4.3 adds a new route set to the network. The following are the parameters for ADD-RTSET:

RTSET—A user-defined name that identifies the route set. Valid values are alphanumeric. Valid lengths are one to 11 characters.

DPC—The destination point code for this route set.

ROUTE1 and **ROUTE2**—Specifies the link sets in this route set. Note that the link sets should be listed in priority order.

NOTE: A default route set is automatically created by the system for each link set defined in MTP Level 2 Provisioning. The system creates the default route set name by appending the link set name with either STP (signaling transfer point) or SEP (signaling end point). In this example, the default route set names are LSET1STP and LSET2STP.

The **ADD-ROUTE** commands in lines 14 and 15 of Figure 4.3 add the default route sets for the link sets defined in the MTP 2 Level Provisioning part. The default route sets must always be added to the MTP 3 Level Provisioning part of the configuration file.

The following are the parameters for ADD-ROUTE:

RTSET—The name of the default route set for the link set. You defined the link sets in MTP2 Provisioning, lines 3 and 4 in Figure 4.2. The system created the name for the default route set (RTSET) by appending the link set name with either STP or SEP. For example, in Figure 4.2, the link sets are LSET1 and LSET2. Therefore, the default route set names (RSET) are LSET1STP and LSET2STP respectively.

LSET—The name of the link sets you defined in MTP2 Provisioning (lines 3 and 4 in Figure 4.2).

PRIORITY—The priority of the link set. Valid values for PRIORITY when specified in the RTSET command are 0 through 3, where 0 is the highest priority and 3 is the lowest.

The default route sets have their associated link sets as the first priority route (0). Therefore, any routes you add to the configuration must start with priority = 1.

The **MODIFY-LSET** commands in lines 16 and 17 sets the link sets to active. The following are the parameters for MODIFY-LSET:

LSET—A user defined value that identifies the link set. Valid values are alphanumeric.Valid lengths are 1 to 8 characters.

ADMINSTATE—Sets the administrative state of the link set. Valid values are ACTIVE (activate link set) or INACTIVE (deactivate link set).

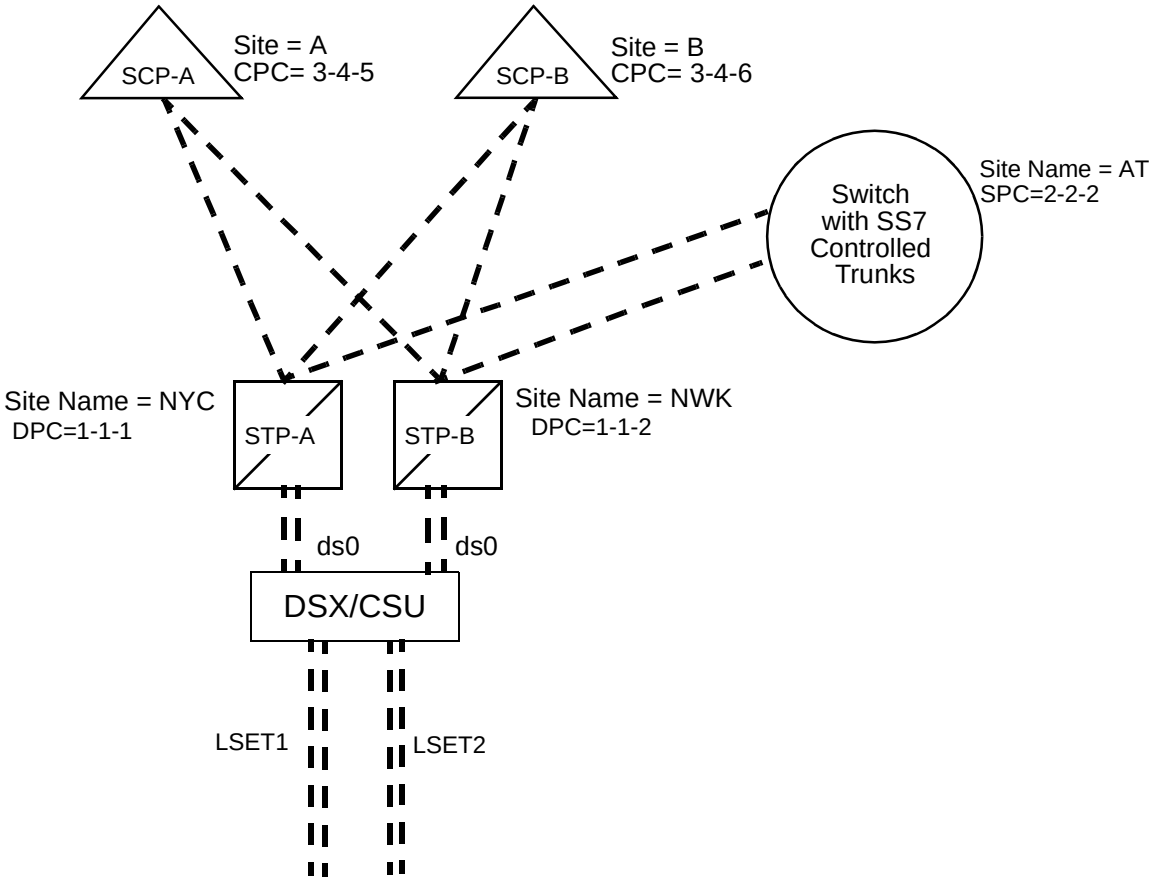
The **MODIFY-ALARM-CONFIG** command in line 18 allows the user to display alarm messages on the console. The following are the parameters for MODIFY-ALARM-CONFIG:

DISPLAY—Enables or disables the displaying of alarm messages. Valid values are ON and OFF.

Line 19 exits the MML program.

4.4.2.3 TCAP Provisioning

The TCAP Provisioning part of the MTP configuration file defines the SCCP configuration used for TCAP message routing. Figure 4.4 is an example of this part.



Lines in **<filename>**, where filename is recommended to be called **tcap.mml** (or any name tha includes tcap and has the extension .mml) for TCAP Provisioning.

(1) ADD-SNSP:SPC=1-1-1;
(2) ADD-SNSP:SPC=1-1-2;
(3) ADD-SUBSYS:SPC=1-1-1,SSN=50;
(4) ADD-SUBSYS:SPC=1-1-2,SSN=51;
(5) ADD-CPC:SPC=1-1-1,SSN=20,CPC=3-4-5;
(6) ADD-CPC:SPC=1-1-2,SSN=21,CPC=3-4-6;
(7) EXIT:;

Figure 4.4: TCAP Provisioning Example

The **ADD-SNSP** commands in lines 1 and 2 of Figure 4.4 add new signaling points to the SCCP network. The following are the parameters for ADD-SNSP:

SPC—Signaling point code entered as Network-Cluster-Member (14-bit).

NOTE: The SPC must already be provisioned in the MTP network. When an SPC is added to the SCCP network, the SCCP management subsystem (SSN=1) is automatically created by the SCCP in order to display remote SCCP status in ITU WHITEBOOK networks. When a remote user part (SCCP) is unavailable, only one SST message is sent to the remote SCCP for SSN=1 until the remote SCCP is up. Subsystem SSN=1 can only be displayed by users to monitor the remote SCCP's status. It cannot be modified by users.

The **ADD-SUBSYS** commands in lines 3 and 4 of Figure 4.4 add new subsystems to the SPCs defined in lines 1 and 2. The following are the parameters for ADD-SUBSYS:

SPC—Signaling point code entered as Network-Cluster-Member (14-bit).

SSN—A subsystem number with a range of 2 to 255.

The **ADD-CPC** commands in lines 5 and 6 of Figure 4.4 add new concerned point code (CPC) to the identified subsystems (lines 3 and 4) of the defined SPCs (lines 1 and 2). The following are the parameters for ADD-SNSP:

SPC—Signaling point code entered as Network-Cluster-Member (14-bit).

SSN—A subsystem number with a range of 2 to 255.

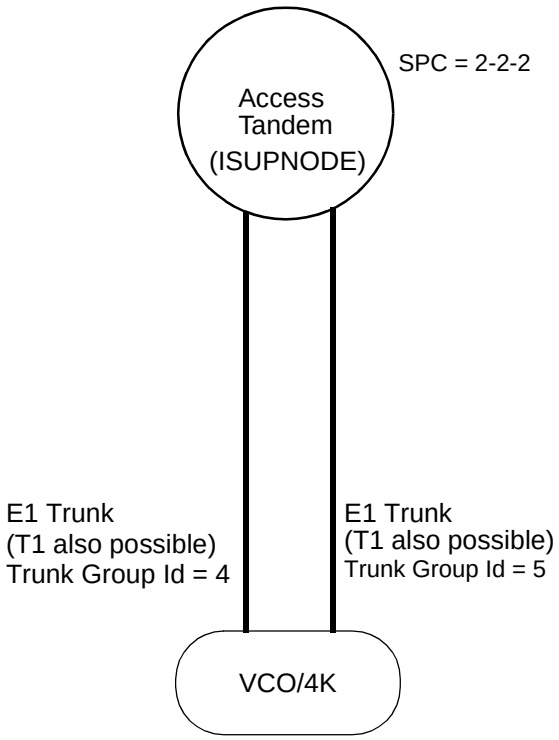
CPC—Concerned point code entered as Network-Cluster-Member (14-bit).

The ADD-CPC call succeeds only if the SP and the subsystem exist in the SCCP network database.

Line 7 exits the MML program.

4.4.2.4 ISUP Level Provisioning Configuration File

The ISUP Level Provisioning configuration file defines the circuits. Figure 4.5 is an ISUP provisioning example for a system with Single Span E1 cards.



Lines in <**filename>**, where filename is recommended to be called **isup.mml** (or any name that includes isup and has the extension .mml) for ISUP Provisioning.

(1) MODIFY-ISUPCONF:CFGNAME=CF0,VARIANT=GENERIC,MNTIND=ON;
(2) ADD-ISUPNODE:PCNO=1,DPC=2-2-2;
(3) ADD-ISUPCGRP:PCNO=1,GRPID=1,CCTNUM=32,TRNKGRPID=4;
(4) ADD-ISUPCCT:PCNO=1,GRPID=1,CCTNUM=0,RANGE=32;
(5) ADD-ISUPCGRP:PCNO=1,GRPID=2,CCTNUM=32,TRNKGRPID=5;
(6) ADD-ISUPCCT:PCNO=1,GRPID=2,CCTNUM=0,RANGE=32;
(7) MODIFY-ISUPOFFINFO:PCNO=1,ANMOFF=1,ACMOFF=1,CRGOFF=0;
(8) DISPLAY-ISUPNODE:PCNO=1,DPC=2-2-2;
(9) DISPLAY-ISUPCGRP:PCNO=1,GRPID=*;
(10) DISPLAY-ISUPCCT:PCNO=1,GRPID=1,CCTNUM=*;
(11) DISPLAY-ISUPCCT:PCNO=1,GRPID=2,CCTNUM=*;
(12) EXIT:;

Figure 4.5: ISUP Level Provisioning Example

The **MODIFY-ISUPCONF** command in line 1 of Figure 4.5 modifies the current ISUP configuration. **You must add this line for every SP.** The following are parameters for MODIFY-ISUPCONF:

CFGNAME—Specifies the ISUP configuration name. The system names ISUP configuration files using the convention **CF<sp>**, where sp is the signaling point number of the logical node. Valid values are 0 to 7; you *must start with 0* and increment by ones (i.e. 0, 1, 2...).

VARIANT—Specifies the country variant (see *Appendix E* for more information on country variants). The following are the valid values for this parameter:

- GENERIC (ITU White Book; use for Germany/Switzerland)
- Q767 (Chile)
- THAILAND
- HONGKONG
- SINGAPORE
- ITALY
- FINLAND
- AUSTRALIA
- SPAIN

NOTE: Each SP may specify a different variant.

MNTIND—Sets the maintenance indicator. Valid values are ON (maintenance indicators are sent to the Maintenance module) or OFF (maintenance indicators are not sent to the Maintenance module)

NOTE: Set this parameter to ON unless Cisco Systems Technical Support advises you to set it to OFF.

The **ADD-ISUPNODE** command in line 2 adds a signaling node to the ISUP database. The following are the parameters for ADD-ISUPNODE:

PCNO—A unique point code index number that refers to the node's destination point code.

DPC—The destination point code for the signaling node.

The **ADD-ISUPCGRP** command (lines 3 and 5) adds Circuit Groups to the ISUP database. The following are the parameters for ADD-ISUPCGRP:

PCNO—A unique point code index number that refers to the DPC.

GRPID—A value that identifies a circuit group. Most often, circuit groups are set up to correspond to a span of an E1/T1 card.

GRPID is a user-defined value you use as a multiplier when you calculate the circuit identification codes (CICs). GRPIDs, along with the DPC, identify specific circuits to the SS7 network. **Valid values are decimal, 0 through 127. Valid values for Australia are decimal, 1 through 127. Each GRPID must be unique per PCNO.**

TIP: *The circuit identification code (CIC) is a decimal number that identifies each circuit to the SS7 network. Calculate the CIC by using the following expression:*

CIC = (GRPID x 32) + Circuit Number (valid values are 0 to 31)

Calculate Australian CICs by using the following expression:

CIC = [(GRPID-1) x 31] + Circuit Number (valid values are 1 to 31)

Since GRPID can equal 0 through 127 (1 through 127 for Australia) and ITU circuit numbers can equal 0 to 31 (1 to 31 for Australia), ITU CICs can equal 1 through 4,095 for non-Australia, or 1 through 3,937 for Australia.

Max_CIC = (127 x 32) + 31 = 4,095

or

Max_CIC = (126 x 31) + 31 = 3,937 for Australia

CICs are not unique and can be duplicated for different signaling nodes (PCNO).

CCTNUM—Specifies both maximum number of circuits and limits the circuit numbers available to this group (see ISUPCCT example on next page). Values are from 1 to 32 (1 to 31 for Australia).

NOTE: Cisco Systems recommends you set the default value to 32.

TRNKGRPID—A unique value that identifies a circuit group. Most often, circuit groups are set up to correspond to a span of an E1/T1 card.

TRNKGRPID is a unique, user-defined value you use as a multiplier when you calculate the global circuit identification codes (GCICs). TRNKGRPIDs identify specific circuits to Circuit Interworking. **Valid values are decimal, 1 through 75 in 2K Mode; 1 through 171 in 4K Mode.**

NOTE: You cannot share trunk group circuits between SPs.

TIP: *The global circuit identification code (GCIC) uniquely identifies every circuit in the SS7 subsystem domain. GCICs use the Trunk ID (TRNKGRPID) as a multiplier in the calculation. Calculate GCICs using the following expression:*

GCIC = (TRNKGRPID x 32) + Circuit Number (valid values are 0 to 31)

Calculate Australian GCICs using the following expression:

GCIC = [(TRNKGRPID-1) x 31] + Circuit Number (valid values are 1 to 31)

Lines 4 and 6 add the individual circuits to each ISUP circuit group through **ADD-ISUPCCT**. The following are the parameters for this command:

PCNO—A unique point code index number that refers to the DPC for the ISUP signaling node.

GRPID—Specifies the ID of the circuit group for this set of circuits. This is the same GRPID as GRPID in lines 3 and 5.

CCTNUM—Specifies the first circuit number in this group, from 0 to 31 (1 to 31 for Australia). **Do not confuse this parameter with CCTNUM in lines 3 and 5.**

RANGE—Specifies the range of sequential numbers used to identify the circuits. Valid values are 1 through 32.

*NOTE: CCTNUM plus RANGE **cannot** exceed the previously set ADD-ISUPCGRP CCTNUM value.*

For example, a CCTNUM of 0 and a RANGE of 32 creates 32 circuits, numbered 0 through 31 (CCTNUM + RANGE = 32, therefore, the previous ADD-ISUPCGROUP CCTNUM would have to be at least 32). Likewise, a CCTNUM of 10 and a RANGE of 16 creates 16 circuits, numbered 10 through 25 (CCTNUM + RANGE = 26, therefore, the previous ADD-ISUPCGROUP CCTNUM would have to be at least 26).

The **MODIFY-ISUPOFFINFO** command in line 7 modifies the ISUP office information in the ISUP database. The following are the parameters for MODIFY-ISUPOFFINFO:

PCNO—A unique point code index number that refers to the DPC for the ISUP signaling node.

ANMOFF—Sets answer message office. Valid values: 0 (OFF), indicating this is not the office type; or 1 (ON), indicating this is the office type. Setting to 0 (OFF) disables the T9 ISUP timer.

ACMOFF—Sets address complete office. Valid values: 0 (OFF), indicating this is not the office type; or 1 (ON), indicating this is the office type. Setting to 0 (OFF) disables the T7 ISUP timer.

CRGOFF—Sets charge office. Valid values: 0 (OFF), indicating this is not the office type; or 1 (ON), indicating this is the office type.

Note: The default value is 1 (was 0 in earlier versions of EBS). Please make sure you set this to 0 if your configuration requires it.

Lines 8 to 11 of the example define the ISUP reports to DISPLAY:

- Line 8 sets up the ISUP node report for display.
- Line 9 sets up the circuit group report for display.
- Lines 10 and 11 set up the report for circuit groups 1 and 2 for display.

Line 12 exits the MML program.

Refer to the ISUP Configuration section in the *NewNet AccessManager Manual* for descriptions and examples of these reports.

4.4.3 Creating/Modifying and Loading the Configuration Files for ISUP/TCAP

This subsection describes how to create or modify the MTP, ISUP, and TCAP configuration files, and how to load them to run ISUP applications. This procedure only needs to be run once.

To create or modify the configuration files, complete the following steps:

1. If you haven't already done so, log in to the SS7 subsystem as **cktint**.
2. Use a standard UNIX editor to create the MTP configuration files for your site.

   NOTE: Lines in each configuration file must be ordered as shown in this subsection.
3. If the EBS stack is running, enter the following command and press **Return**:

   stop-ss7.sh

   Type **px** and press **Return** to verify no processes are running.
4. Enter **rmdb 0** and press **Return**. This deletes all of the files for SP 0 in the EBS database. Continue entering **rmdb <n>** and pressing **Return** for each additional SP.

   NOTE: If you wish to save the existing configuration files, you should copy these files to another location instead of deleting them.

   NOTE: To modify a specific SP at a later date, you must first run an **rmdb <n>** *command to clear that SP's files in the EBS database.*
5. Enter the following command and press **Return**:

   ebs_start
6. **cd** to **~/sys/SPcc**. Each directory after SP 0 is **~/sys/SPcc<n>**.
7. Enter the following command and press **Return**:

   upmd **<n>** &
8. Enter the following command and press **Return**:

   snmd **<n>** &

9. Configure MTP Level 2/3.

   Take the .mml file of commands in a single statement:

   mml **<n>** <**filename**>

   where **n** is the SP number and **filename** is what you created in MTP Level 2/3 Provisioning (See *Section 4.4.2.1* and *Section 4.4.2.2*). *In a single-SP configuration,* ***n*** *is always 0.*

   If you use the single command line method, make sure that the MTP level configuration file does not contain blank or comment lines and ends with EXIT:;

   As an alternative, you can cut and paste the MTP-related command lines from your MTP configuration file, one-by-one, to the command line of the MML program. Refer to Figure 4.2 and Figure 4.3 for examples of MTP-related commands.

   NOTE: Cisco Systems recommends the single command line method.

10. You will receive a <SUCCESS> message for each correct line/command in the .mml file. If everything is correct, go on to Step 13.

    If you receive an <ERROR> message, count down to which line the message represents and investigate/edit the original file. Make changes to correct the problem (i.e. typos, etc.), then stop the EBS stack by entering the following command and pressing **Return**:

    stop-ss7.sh

    Type **px** and press **Return** to verify no processes are running.

    Restart at Step 5.

11. If you are using TCAP, do Step 11 and Step 12. If not, go on to Step 13.

    Enter the following command and press **Return**:

    scmd 0 &

12. Configure TCAP Level.

    Take the .mml file of commands in a single statement:

    mml 0 <**filename**>

    where **filename** is what you created in TCAP Level Provisioning (See *Section 4.4.2.3*).

    If you use the single command line method, make sure that the TCAP level configuration file does not contain blank or comment lines and ends with EXIT:;

    As an alternative, you can cut and paste the TCAP-related command lines from your TCAP configuration file, one-by-one, to the command line of the MML program. Refer to Figure 4.4 for examples of TCAP-related commands.

    NOTE: Cisco Systems recommends the single command line method.

13. Enter the following command and press **Return.**

    AccessISUP 0 &

14. Configure ISUP Level.

    Take the .mml file of commands in a single statement:

    mml 0 <**filename**>

    where **filename** is what you created in ISUP Level Provisioning (See *Section 4.4.2.4*).

    If you use the single command line method, make sure that the ISUP configuration file does not contain blank or comment lines and ends with EXIT:;

    As an alternative, you can cut and paste the ISUP-related command lines, one-by-one, to the command line of the MML program. Refer to Figure 4.5 for examples of ISUP-related commands.

    NOTE: Cisco Systems recommends the single command line method.

15. Repeat Step 6 through Step 14 for each additional SP, changing <n> to the appropriate signaling point (i.e., upmd 0-7, snmd 0-7, mml 0-7 <MTP mml file>, AccessISUP 0-7 and mml 0-7 <ISUP mml file>).

16. You will receive a <SUCCESS> message for each correct line/command in the .mml file. If everything is correct, go on to Step 17.

    If you receive an <ERROR> message, count down to which line the message represents and investigate/edit the original file. Make changes to correct the problem (i.e. typos, etc.), then stop the EBS stack by entering the following command and pressing **Return**:

    stop-ss7.sh

    Type **px** and press **Return** to verify no processes are running.

    Restart at Step 5.

17. Stop the EBS stack by entering the following command and pressing **Return**:

    stop-ss7.sh

    Type **px** and press **Return** to verify no processes are running.

18. Restart EBS to load the new configuration file, as described in *Section 5.2*.

    *NOTE: **Do not** start CktInt until you configure as shown in Section 4.5.1.*

19. If using TCAP, start SEPT as described in *Section 5.3*.

4.5 PLATFORM CONFIGURATION AND RESOURCE PROVISIONING

This subsection describes how to create the platform configuration and resource provisioning files.

4.5.1 Cktint Configuration File: CktInt.cfg

The cktint configuration file specifies the data necessary for the Circuit Interworking software to interface with the VCO/4K system and the host computer. You can modify the default configuration file, **CktInt.cfg**, to suit your requirements. **CktInt.cfg** resides in directory **~/sys/CktintAnEnv ($XNV)**.

Figure 4.6 is a sample of the **CktInt.cfg** file for a redundant configuration with four hosts.

```
*-HOST_LINK01:host-1.6050
  -HOST_LINK02:host-2.6060
  -HOST_LINK03:host-3.6070
  -HOST_LINK04:host-4.6080
*-SDS_LINK01:vcoa.7010
  -SDS_LINK02:vcoa.7020
  -SDS_LINK03:vcoa.7030
  -SDS_LINK04:vcoa.7040
  -HOSTLOAD:ROUNDROBIN [default: BROADCAST]
  -OUT_COT_ORULE2
  -OUT_COT_HZ2010
  -DEBUG2
  -DEBUG5
  -DEBUG9
  -DEBUG11
  -DEBUG12
  -DEBUG21
  -DEBUG22
```

NOTE: Cisco Systems recommends only setting debugs for troubleshooting or testing.

* Mandatory settings

Figure 4.6: Sample CktInt.cfg File

Table 4.1 lists and describes the configuration parameters in CktInt.cfg.

Table 4.1: CktInt.cfg Parameters

Parameters	Description
-HOST_LINK0n:hostname.tcp_port *(mandatory-at least one host link **must** be defined)*	Specifies the logical TCP link(s) between the host(s) and the SS7 subsystem. The following are the variables for -HOST_LINK0: **n—**The link number. Valid values are 1 to 8. **hostname**—Any valid host name (e.g. localhost). **tcp_port**—The decimal value of the TCP port logical port address (socket number) on the SS7 subsystem to which the host connects. For more information on host and VCO/4K links, refer to *Section 1.3*.
-SDS_LINK0n:vconame.tcp_port *(mandatory-at least one VCO/4K link **must** be defined)*	Specifies the logical TCP link(s) between the SS7 subsystem and The VCO/4K system. The following are the variables for -SDS_LINK0: **n**—The link number. Valid values are 1 to 8. **vconame**—The name of the VCO/4K the SS7 system is using. **tcp_port**—The decimal value of the TCP port logical port address (socket number) on the VCO/4K to which the SS7 subsystem connects. There must be a one-to-one correspondence between these links and the links specified by -HOST_LINK0. For more information on host and VCO/4K links, refer to *Section 1.3*.
-HOSTLOAD:mode *(optional)*	Specifies the SS7 call load sharing mode for the hosts. Valid values are BROADCAST or ROUNDROBIN. The default is BROADCAST. For more information on load sharing modes, refer to *Section 1.3*.
-CKTMAP**filename** *(optional)*	Where the user defines the **filename**. Specifies the circuit configuration file residing in the **$XNV** directory; contains the address translation parameters for SS7 circuit to VCO/4K ports. If this flag is not defined, the default file is **$XNV/ckt_ss7_to_sds**.

Table 4.1: CktInt.cfg Parameters (Continued)

Parameters	Description
-GRPMAP**filename** *(optional)*	Where the user defines the **filename**. Specifies the group configuration file residing in the **$XNV** directory; contains the parameters for each trunk group. If this flag is not defined, the default file is **$XNV/grp_ss7_to_sds**.
-PRMORD**filename** *(optional)* *(not available for multiple-SP configurations)*	Where the user defines the **filename**. Specifies the configuration file residing in the **$XNV** directory; controls the order of SS7 message parameters. The format of each line in the file is the message type followed by the ordered list of parameter types. There is a separate file for each supported variant. If this flag is not defined, the default file is **$XNV/param_ord.cfg**. If you have a multiple-SP configuration, filenames are fixed and you cannot change their definition.
-OUT_COT_ORULE**rule** *(optional)* -OUT_COT_HZ**hz** *(optional)*	Supports outbound COT; **rule** is the outpulse rule you set up for COT and **hz** is either 2010 or 1780 hertz. To use outbound COT messages, add these two parameters to the end of your **CktInt.cfg** file and set up the following outpulse rule and supervision template: Outpulse Rule: TIME SUP 5 FINAL SUP # Supervision Template #: TIME OK ISUP Tone OKREP Refer to *Section 6.8.1* for outbound COT call flow examples. *NOTE: Australia does not support outbound COT.*
-RESGRP**filename** *(optional)*	Where the user defines the **filename**. Specifies the resource configuration file residing in the **$XNV** directory; controls Rotary/Cyclic port selection. If this flag is not defined, the default file is **$XNV/res_grp.cfg**. The default behavior is to hunt resource groups in Cyclic mode. Refer to *Section 4.5.1.1* for more information on this feature.

Table 4.1: CktInt.cfg Parameters (Continued)

Parameters	Description
-AOCTRANSIT *(required if the exchange is transit for Advice of Charge)*	If Feature Flag 14 is turned on to enable AOC functionality and the exchange is a transit exchange for AOC, this line must also be added. Otherwise, the exchange will be considered an originating or terminating exchange for AOC. *NOTE: Variant must be set to "Generic" in ISUP Level Provisioning.*
-DEBUGn (optional—use for troubleshooting)	n specifies the debug flag number, which must be set. The following are valid values for -DEBUGn:
	1 - High level interworking message debug
	*2 - VCO/4K Message Trace
	3 - IPC key trace
	4 - Displays switchover state machine transitions
	*5 - Host message trace
	6 - Used for detecting inability to find circuits or ports
	*9 - Message trace for non-SS7 calls
	*11 - Trace of transmitted SS7 messages
	*12 - Prints xmitted SS7 messages in detail; *11 must be on*
	13 - Displays the **parm_order** search
	14 - Displays the undefined SS7 parameter types that were transmitted
	**15 - Traces the $DC reports received from the VCO/4K at switchover
	16 - Prints related diagnostic messages in the cktint log file concerning the Host Control of Call Load feature
	20 - Reserved for maintenance interface
	*21 - Trace of received SS7 messages
	*22 - Prints received SS7 messages in detail; *21 must be on*
	24 - Displays the undefined SS7 parameter types that were received
	25 - Trace of circuit state from AccessISUP (AI) layer

Table 4.1: CktInt.cfg Parameters (Continued)

Parameters	Description
-DEBUGn (cont.)	26 - Circuit query response
	27 - Generated SS7 primitives trace
	28 - Continuity check procedures debug
	29 - Forces circuits to come into service ACTIVE_IDLE
	30 - Call processing debug
	31 - Debugs reading templates into objects
	32 - Debugs building SS7 messages from templates
	33 - Prints detailed information in the log file concerning invalid SS7 messages
	34 - Prints detailed information in the log file concerning protocol violations
	35 - Prints detailed information in the log file concerning IAM message which were received in an invalid circuit state
	36 - Prints detailed information in the log file concerning call object creation and deletion
	40 - Turns off IAM invalid parameter checking
	41 - Prints detailed information in the log file concerning Advice of Charge (AOC) messages
	* Standard for troubleshooting ** Standard for troubleshooting redundant systems only
	NOTE: If multiple debug flags are needed, this parameter must be specified multiple times.
	*TIP: Type **dbs** to display debug flags that are turned ON.*
-FEATURE_FLAGn *(optional)*	n specifies the feature flag number, which must be set. The following are valid values for -FEATURE_FLAGn:
	01 - Host receives SS7 $EA reports for any RELs that it sends.
	02 - Includes End of Optional Parameter (EOP) in SS7 $EA reports.

Table 4.1: CktInt.cfg Parameters (Continued)

Parameters	Description
-FEATURE_FLAGn (cont.)	04 - Specifies a one-byte port address in standard mode. *NOTE: Not compatible with extended operational mode.*
	05 - Enables the Host Control Option. Allows only one host to control a call. A secondary host can assume control of a controlling host's calls by sending an SS7 $C0 05 command.
	06 - Causes MTP link alarms to be processed by cktint and an SS7 $F0 alarm condition report to be sent to the host.
	07 - An incoming CGB/CGU/GRS from the network, system, or isup_console generates an SS7 $D9 Circuit Group State report to the host instead of individual $D3 System Port Status reports.
	08 - Host message queue is not flushed when socket is dropped.
	09 - Does not drop host socket connection if the socket write is blocked.
	10 - Protocol violation messages suppressed from log file.
	11 - Ignores the inpulse rule number in ckt_ss7_to_sds file.
	12 - When a $70 command gets rejected with "Invalid Class of Service," cktint releases a call, if one exists, and sends an $EA REL report to the host and a $70 on-hook command to the VCO/4K.
	13 - Disables cktint from sending automatic RLCs when a REL message is received from the network. Instead, the host must send the RLC. *Will not work if the incoming and outgoing circuits are associated in the $49 command.*
	14 - Enables Advice of Charge (AOC) functionality. *NOTE: Variant must be set to "Generic" in ISUP Level Provisioning.*

Table 4.1: CktInt.cfg Parameters (Continued)

Parameters	Description
-FEATURE_FLAGn (cont.)	15 - When the load sharing mode is set to BROADCAST, allows cktint to broadcast a $C0 05 relinquish control command acknowledgement from the VCO to all connected hosts.
	16 - Generates $D3/$D9 reports for maintenance messages like Circuit Group Blocking Acknowledgement (CGBA) or Blocking Acknowledgement (BLA) from the network.
	17 - Informs the host that an outgoing call is being cleared for dual seizure conditions. An $EA report with cause value 0x7E is sent to the host from cktint for a Release (REL) during dual seizure conditions.

4.5.1.1 Rotary/Cyclic Port Selection

This parameter allows you to specify the mode that the system uses for hunting and allocating a circuit for a call.

In CYCLIC mode, the system selects the circuits from the resource group in a sequential manner; the next IDLE circuit in the group is selected for the new call. After the system has selected the last circuit in the sequence, it begins hunting from the first circuit in the group.

In ROTARY mode, the system always begins hunting sequentially from circuit # 1. The first IDLE circuit in the sequence is always selected for the new call.

You can modify the default resource group file, called **res_grp.cfg** in the **$XNV** directory, to suit the needs of your installation with any UNIX text editor. You can rename the file, but the renamed file must still reside in the **$XNV** directory for this feature to work. Figure 4.7 is an example of a **res_grp.cfg** file with 12 resource groups configured.

Resource Group	VCO/4K Device	Circuit Selection Mode
1	0	ROTARY
2	0	ROTARY
3	0	ROTARY
4	0	ROTARY
5	0	CYCLIC
6	0	CYCLIC
7	0	CYCLIC
8	0	ROTARY
9	0	ROTARY
10	0	ROTARY
11	0	ROTARY
12	0	CYCLIC

Figure 4.7: Resource Group Configuration File Example

Table 4.2 summarizes the fields in the resource group configuration file.

Table 4.2: Resource Group Configuration File Fields

Field	Definition
Resource Group	You can define from 1 to 63 resource groups in standard mode and 1 to 223 resource groups in extended 4K mode.
VCO/4K Device	Reserved. Always 0.
Selection Mode	CYCLIC or ROTARY. This mode applies to all circuits in the resource group.

NOTE: The default behavior of the res_grp.cfg file is to hunt resource groups in CYCLIC mode.

4.5.2 Resource Provisioning Files: ckt_ss7_to_sds & grp_ss7_to_sds

The configuration files **ckt_ss7_to_sds** and **grp_ss7_to_sds** contain the parameters for provisioning the circuit IDs and the circuit groups. The circuit configuration file **ckt_ss7_to_sds** defines parameters for SS7 circuits. The circuit group configuration **grp_ss7_to_sds** file defines parameters for circuit groups. Sample **ckt_ss7_to_sds** and **grp_ss7_to_sds** files already exist in directory **~/sys/CktintAnEnv**. You can modify the sample files to suit the needs of your installation.

4.5.2.1 ckt_ss7_to_sds

The **ckt_ss7_to_sds** file contains the address translation parameters for SS7 circuit to VCO/4K ports. Each line in the file corresponds to a single circuit. Figure 4.8 is an example of a **ckt_ss7_to_sds** file for a configuration with two Single Span E1 cards.

> *NOTE:* ***CCTNUM is a hex value in the ckt_ss7_to_sds file.*** *In other configuration files (i.e., .mml configuration file) CCTNUM is a decimal value.*

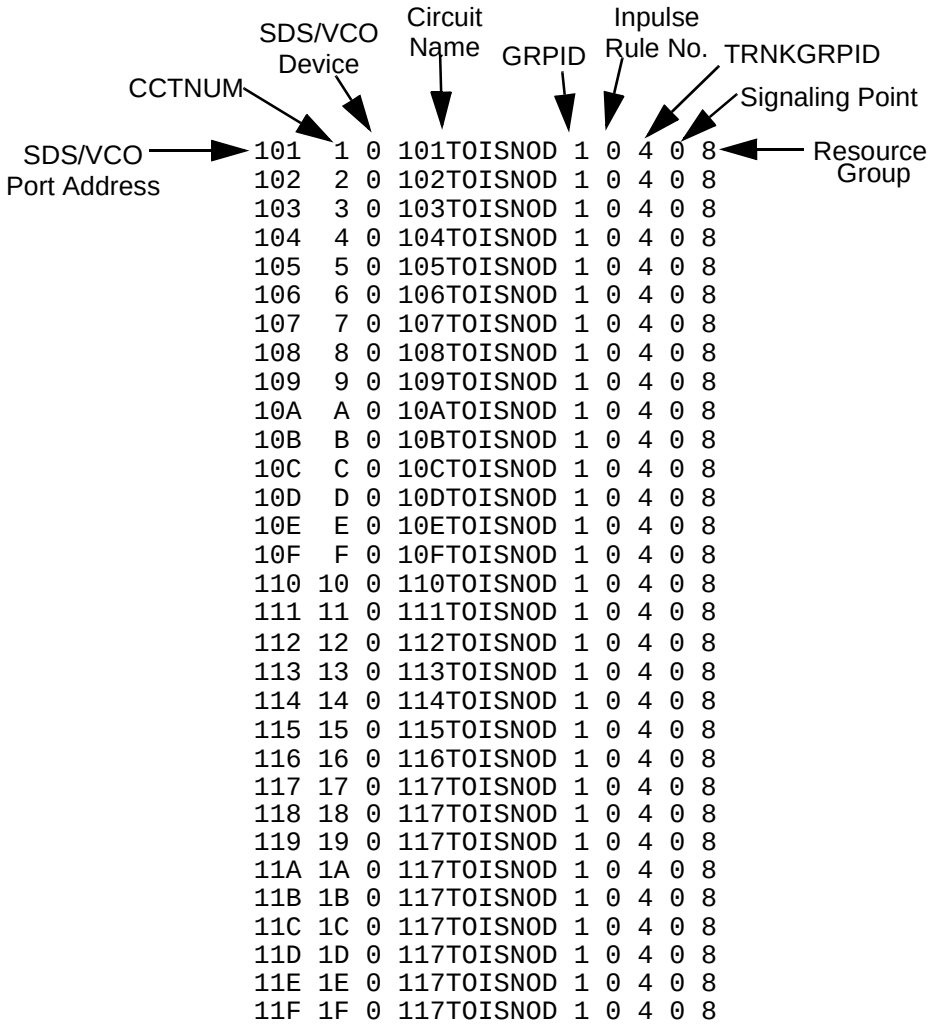


Figure 4.8: Sample ckt_ss7_to_sds Configuration File

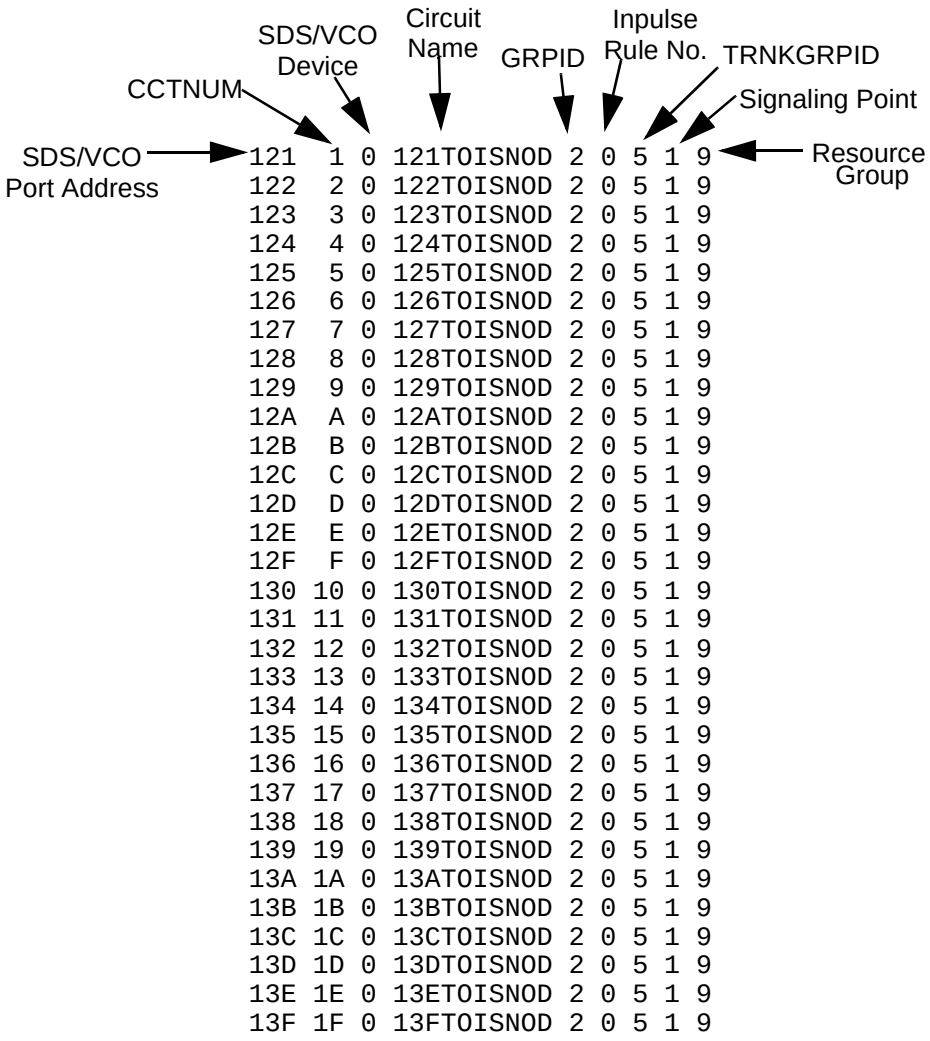


Figure 4.8 (Continued): Sample ckt_ss7_to_sds Configuration File

Table 4.3 lists and describes the fields shown in Figure 4.8.

Table 4.3: Circuit Configuration Fields

Configuration Field	Description
VCO/4K Port Address	Hex value representing the VCO/4K port address.
CCTNUM	Circuit number. Hexadecimal value. Use the same value as in the EBS SS7 stack configuration file, but convert to a hexadecimal number. See Figure 4.5.
VCO/4K Device	Reserved. Set to zero (0).
Circuit Name	40-character ASCII string.
GRPID	Circuit Group ID. Use the same value as in the EBS SS7 stack configuration file (see Figure 4.5). **Valid values are decimal, 0 through 127 or 1 through 127 for Australia.**
Inpulse Rule Number	Reserved. Set to zero (0).
TRNKGRPID	Trunk Group ID. Use the same value as in the EBS SS7 stack configuration file (see Figure 4.5). **Valid values are decimal, 1 through 75 in 2K Mode; 1 through 171 in 4K Mode.** *NOTE: You cannot share trunk group circuits between SPs.*
Signaling Point	The signaling point for this circuit. Valid values are 0 to 7.
Resource Group	You can define from 1 to 63 resource groups in standard mode and 1 to 223 resource groups in extended 4K mode.

4.5.2.2 grp_ss7_to_sds

Each line in the **grp_ss7_to_sds** file specifies parameters for a single trunk group. You must add a space and the 12-character string for the Common Language Location ID CLLI code after each circuit group. An example line from a **grp_ss7_to_sds** file is shown in Figure 4.9.

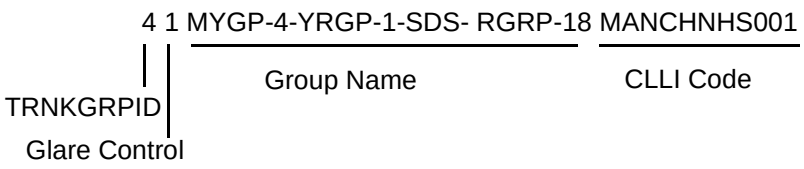


Figure 4.9: Sample Line From grp_ss7_to_sds File

Table 4.4 lists and describes the fields in the **grp_ss7_to_sds** file.

Table 4.4: Circuit Group Configuration Fields

Configuration Field	Description
TRNKGRPID	Trunk Group ID. Use the same value as in the EBS SS7 Stack configuration file (see Figure 4.5). **Valid values are decimal, 1 through 75 in 2K Mode; 1 through 171 in 4K Mode.** *NOTE: You cannot share trunk group circuits between SPs.*
Glare Control	Use for double seizing control indicator in Circuit Group Characteristics Indicator parameter. The following are the valid values for Glare Control: 0 - Control no circuits. 1 - Control odd circuits. 2 - Control even circuits. 3 - Control all circuits. *NOTE: Glare control configuration setting in CktInt must match EBS glare configuration setting for ALL, NONE, EVEN, and ODD.* *If EBS glare configuration is set to "DEFAULT," CktInt glare should be set to ODD if the local/VCO4K point code is lower than the remote point code.Otherwise, CktInt glare should be set to EVEN if the local/VCO4K point code is higher than the distant end point code.* *For networks where the VCO4K is connected to more than one remote point code, it is possible to have a mix of both EVEN and ODD glare control if the distant end point codes are both lower and higher, respectively, than the local VCO4K point code.*
Group Name	40-character ASCII group name
CLLI Code *	Common Language Location ID code. 12-character string.

* Bytes 1-5 are the town code, bytes 6 and 7 are the state code, bytes 8 and 9 are building codes, bytes 10 and 12 are subdivision codes.

4.5.3 SEPT Configuration File: SeptCcDflt.cfg

The SEPT (Signaling End Point Translator) configuration file specifies the data necessary for the SEPT portion of the TCAP software to interface with the SS7 stacks and the host computer. You must modify the default configuration file, **SeptCcDflt.cfg**, to suit your requirements. **SeptCcDflt.cfg** resides in directory **~/sys/SeptTcEnv ($SNV)**.

Figure 4.10 is a sample of the **SeptCcDflt.cfg** file.

```
  -k32
*-d0.140.6
*-r5
*-o20
 -s0
 -ilocalhost
 *-P6050
  -m2
 *-tFileHoldingTempfileNames
 -v
 -x
```

* You must modify these to reflect your specific system configuration.

Figure 4.10: Sample SeptCcDflt.cfg File

Table 4.5 lists and describes the configuration parameters in SeptCcDflt.cfg.

Table 4.5: SeptCcDflt.cfg Parameters

Parameters	Description
-k**key value** *(optional)*	Where the user defines **key value**. Specifies the key value used by SEPT for creating Unix shared memory resources. Keys should be assigned in increments of 16 (i.e. 16, 32, 48...); 0 is not a valid value. The default value is 32.
-d**DPC** *(mandatory)*	Where the user defines **DPC** (Destination Point Code). Specifies which DPC is sent to EBS unless the host indicates a different value in its message. You must modify this parameter to reflect your specific system configuration.
-r**SSN** *(mandatory)*	Where the user defines **SSN** (Destination Subsystem Number). Specifies which SSN is sent to EBS unless the host indicates a different value in its message. You must modify this parameter to reflect your specific system configuration.
-o**LocalSubystemNumber** *(mandatory)*	Where the user defines **LocalSubystemNumber**. You must modify this parameter to reflect your specific system configuration.
-s0 *(optional)*	Specifies EBS's SP number – always 0.
-i**localhost** *(mandatory)*	Where **localhost** specifies the host name residing in the **/etc/hosts** file. The host application uses this to decide which is the local node ip address.
-P**LocalPortNumber** *(mandatory)*	Where the user defines **LocalPortNumber.** Specifies the local TCP Port. You must modify this parameter to reflect your specific system configuration.
-m2 *(optional)*	Specifies the communication mode with the host. The only supported value is "2" – SEPT is the TCP server and the host is the TCP Client.
-t**FileHoldingTempfileNames** *(mandatory)*	Where **FileHoldingTempfileNames** refers to the user-created template database file which contains the names of executable TCAP template files. This file is located under the **$SNV/TcTemps** directory. To use this field, you must enter the correct file name that reflects your specific system configuration.
-v *(optional)*	Verbose Mode – prints detailed messages regarding the program during SEPT initialization.
-x *(optional)*	Debug Mode – prints debug messages during program execution. *NOTE: Cisco Systems recommends only setting debug mode for troubleshooting or testing.*

4.6 REDUNDANCY CONFIGURATION

NOTE: In order for redundancy to operate properly, the same version of the SS7 subsystem software must be installed on both sides of the system.

NOTE: A host must be connected to both the active and standby sides of the SS7 system for reliable redundancy operation.

1. Configure the VCO/4K Feature Flags for Redundancy:
   a. Log into the VCO/4K System Administrator Console side A.
   b. From the Main Menu, select System Configuration Menu. From the System Configuration Menu, select System Feature Configuration. The System Features Display screen appears.
   c. Set the following flags on the System Features Display:
      - Redundant System: **Y**
      - Revert to Basic Redundancy: **N**

   For more information, refer to the *System Administrator's Guide.*
2. Log in as **root** to the SS7 subsystem side A, **<hostname>a**.
3. Enter the following command and press **Return**:

   chmod 644 $HOME/.cshrc
4. Use the vi editor to open the file **$HOME/.cshrc** and perform the following steps:
   a. Remove the comment symbol (#) from the following line:

      `#setenv PLTFRMTYP REDUNDANT`

   b. Add the comment symbol (#) to the following line:

      `setenv PLTFRMTYP STANDALONE`

   c. Save the changes and close the file.
5. To have the change take affect (without logging out and logging back in again), enter the following command and press **Return**.

   source $HOME/.cshrc
6. Enter the following command and press **Return**:

   chmod 644 $EBSHOME/access/config/AccessRd.cfg

7. Use the vi editor to open the file **$EBSHOME/access/config/AccessRd.cfg.**
   a. Check the following lines. The lines should read:

      MONITOR_OPTION OFF

      CTS_CONFIGURATION INVERTED

      HOST-A <SS7 subsystem side A hostname>

      (i.e., HOST-A tsup6ss7a)

      HOST-B <SS7 subsystem side B hostname>

      (i.e., HOST-B tsup6ss7b)

   b. If the lines do not match, modify them accordingly.
   c. Save the changes and close the file.
8. Log on as **root** to the SS7 subsystem side B, **<hostname>b**, and repeat Step 3 through Step 7.
9. Check to make sure the SS7 Integrated Software version matches that on side A. Enter the following commands and press **Return** after each:

   %cd $XNV

   %version cktint

10. To bring the SS7 subsystem into operation, go to *Section 5* and start the SS7 stack and Circuit Interworking software.

4.7 2K TO 4K AND 4K TO 2K CONFIGURATOR

4.7.1 2k to 4k Configurator Software

The 2k to 4k Configurator software allows you to configure Circuit Interworking from 2K ports to 4K ports. Running in 4K mode requires an extended host API.

NOTE: To use this option, your VCO must be licensed/configured for 4K ports.

4.7.2 4k to 2k Configurator Software

The 4k to 2k Configurator software allows you to reconfigure Circuit Interworking back to 2K ports after configuring the system for 4K ports. The extended host API option is supported in either mode.

4.7.3 Installation

In order to run the configuration software, you must complete the following steps:

1. If you haven't already done so, install the SS7 subsystem software (V5.0 FSR00 or above) as described in the appropriate system supplement.
2. Log on as **root**.
3. Change directory **cd** to **/export/home**.
4. Enter the following command and press **Return**:

   ```
   /etc/init.d/volmgt stop
   ```

5. Copy the configurator diskette to the system drive:
   a. Insert the configurator diskette (contains 2kto4k and 4kto2k executables) into your storage subsystem's floppy disk drive.
   b. Enter the following command and press **Return**:

      ```
      cpio -icduv -C65536 -I/dev/rfd0
      ```

   c. When the system is finished copying the disk, enter the following command and press **Return**:

      ```
      /etc/init.d/volmgt start
      ```

6. If necessary, change permissions on the executable files by entering the following command and pressing **Return**:

   ```
   chmod +x ./2kto4k ./4kto2k
   ```

7. Run either the **2kto4k** or **4kto2k** executable, depending on the desired configuration.
8. For the changes to take effect, reboot your system by typing **boot** and pressing **Return**.
9. If you have a redundant configuration, repeat Step 5 through Step 8 on the other side.

   NOTE: If Solaris OS is re-installed any time after this process, you must re-copy and re-run the configurator software.

4.7.4 Checksums and Sizes

2k to 4k Configurator Software

File Name	Checksum /usr/bin/sum	Size ls -l
2kto4k	15747 148	75768

4k to 2k Configurator Software

File Name	Checksum /usr/bin/sum	Size ls -l
4kto2k	42158 147	75140

Section 5
SYSTEM ADMINISTRATION

5.1 INTRODUCTION

The SS7 subsystem requires little user interaction during normal day-to-day operation. This section focuses on basic system initialization and monitoring functions that are performed through the UNIX command line. This section is divided into the following areas:

- Starting the SS7 Stack and Circuit Interworking software and configuring for Autostart
- Starting the SEPT software
- Monitoring the SS7 link status and log files
- Managing the circuit and circuit groups
- Initiating a switchover
- Bringing down the SS7 subsystem
- Using scripts/alias

The SS7 subsystem is configured for three users: **root**, **cktint**, and **sept**. Use **cktint** when performing Circuit Interworking functions. Use **sept** when performing SEPT functions.

> *NOTE: The initial password is factory-configured as "abc123" for **root** and **cktint**. However, the password may have been changed when the Circuit Interworking software was installed. If you do not know cktint's password, contact your network administrator.*

5.2 STARTING THE SS7 STACK AND CIRCUIT INTERWORKING

To start the SS7 stack and Circuit Interworking, complete the following steps:

1. Log in as **cktint**.
2. Enter the following command and press **Return**:

   start-ss7.sh

   The following messages appear:

   ```
   This script will assist you in bringing up your Integrated
   SS7 system in a controlled fashion.

   Would you like to start the EBS stack [y/n]?y
   ```

3. Type **y** and press **Return**. Messages similar to the following appear:

   NOTE: The following example reflects a multi-SP configuration with SPs 0 and 1.

```
Starting Signalling Points 0 1
Starting ebs_start...

Signalling Point Manager - Version 3.5
Copyright (c) ADC NewNet, Inc.
All Rights Reserved

Loading /dev/ecp0   - device does not exist
Loading /dev/ecp1   - device does not exist
Loading /dev/ecp2   - device does not exist
Loading /dev/ecp3   - device does not exist
Loading /dev/ecp4   - device does not exist
Loading /dev/ecp5   - device does not exist
Loading /dev/ecp6   - device does not exist
Loading /dev/ecp7   - device does not exist
Loading /dev/ecpt0
Loading /dev/ecpt1  - device does not exist
Loading /dev/ecpt2  - device does not exist
Loading /dev/ecpt3  - device does not exist
Loading /dev/ecpt4  - device does not exist
Loading /dev/ecpt5  - device does not exist
Loading /dev/ecpt6  - device does not exist
Loading /dev/ecpt7  - device does not exist
AccessMANAGER Ready

**************************************
******* AccessALARM is in service *******
*****   Console Output is DISABLED  *****
**************************************

Starting upmd...# :0

User Part Manager - Version 3.5
Copyright (c) ADC NewNet, Inc.
All Rights Reserved
UPMD #0 :Waiting <snmd> to start
Starting snmd...# :0

Signalling Network Manager - Version 3.5
Copyright (c) ADC NewNet, Inc.
All Rights Reserved

UPMD #0 :<snmd>  started, loading prestored data
UPMD #0 :MTP Started
UPMD #0 :EBSRUN is set to /export/home/EBS/access/RUN0
UPMD #0 :Protocol type is CCITT
Starting upmd...# :1

User Part Manager - Version 3.5
Copyright (c) ADC NewNet, Inc.
All Rights Reserved

UPMD #1 :Waiting <snmd> to start
Starting snmd...# :1
```

```
Signalling Network Manager - Version 3.5
Copyright (c) ADC NewNet, Inc.
All Rights Reserved

UPMD #1 :<snmd>  started, loading prestored data
UPMD #1 :MTP Started
UPMD #1 :EBSRUN is set to /export/home/EBS/access/RUN1
UPMD #1 :Protocol type is CCITT
Starting AccessRd...
host_name vt-8ss7a
server_name -vt-8ss7a-  client_name -vt-8ss7b-
filename /export/home/EBS/access/config/AccessRd.cfg
========== INFO READ FROM FILE ============
switch type      = 4 PORT
system_id        = SystemA
serial_port      = /dev/ttyb
pause_timer      = 2000 msec.
state_change_tmr = 20000 msec.
action_order1    = Broadcast_Msg
action_order2    = Start_Script
Monitor_option   = ON
HOST-A nheng-ss7-1a
HOST-B nheng-ss7-16
CTS config       = INVERTED
state_change_max = 5
host_name nheng-ss7-1a
============== STANDBY ==============
status lines 20
    CTS  :0
    DTR  :1
    RTS  :1
REMOTE DEAD
============ HW_STANDBY =============
REMOTE ALIVE
Starting AccessISUP...# :0
Starting AccessISUP...# :1
Starting tli...
server name is tlisrv
wait...
initiating the dispatcher
tli_handler:listening ....
tli_handler:accepting ....
tli_handler:in service ....
SP_NUMBERS :- 0 1
EBS SS7 is now started.

Would you like to start the Circuit Interworking (CktInt) software?
(Note:You should not start CktInt without the EBS stack running!)
[y/n]y
```

4. Type **y** and press **Return**. The following message appears:

```
Starting cktint ...
. . . . .
Circuit Interworking is now started.
```

5.3 STARTING SEPT

To start SEPT, complete the following steps:

1. Start your SS7 stack as described in *Section 5.2*.

   NOTE: Circuit Interworking does not have to be running for SEPT to start.

2. Enter the following command and press **Return**:

   scmd 0 &

3. Log out of **cktint**.
4. Log in as **sept**.
5. Enter the following command and press **Return**:

   start-tcap.sh

   The following messages appear:

   ```
   Starting the SEPT process...
   ```

To stop SEPT, complete the following steps:

1. Log in as **sept**.
2. Enter the following command and press **Return**:

   stop-tcap.sh

   The following message appears:

   ```
   Would you like to stop the SEPT process? [y/n]
   ```

   Type **y** and press **Return** to stop SEPT.

5.4 CONFIGURING CKTINT AND SS7 STACK FOR AUTOSTART

You can configure the system for autostart, so that the SS7 stack and Circuit Interworking are automatically started at system boot. To configure for autostart, either choose the autostart option during CktInt installation, or complete the following steps:

1. Log in as **root**.
2. Enter **cp $BIN/S85ss7 /etc/rc3.d**.

The autostart process takes approximately 20 minutes to execute after SPARC reboot.

*NOTE: Autostart may be configured when running **install_cktint.sh**.*

5.5 MONITORING THE SS7 LINK STATUS

This section describes how to display link status and how to enable SS7 message logging.

5.5.1 Display Link Status

To display the link status, complete the following steps:

1. Enter the following command and press **Return**:

   ```
   ss7-links
   ```

   Figure 5.1 is an example of the of the output from the **DISPLAY-LINK** command. Possible values for STATE are as follows:

   ```
   -A (aligned), which indicates that the link is up and active

   -F (failed), which indicates that the links failed to align and
   that they are down

   -LI (locally inhibited)

   -RI (remotely inhibited)
   ```

```
-----------------------------------------------------------------
     LINK     PORT     LSET    SLC      STATE    ACTIVATED   TYPE
-----------------------------------------------------------------
  STP-MAN-0     1    STP-MAN     0         -F           NO     DTE
 STP-CONC-0     2   STP-CONC     0         -A          YES     DTE
 STP-CONC-1     3   STP-CONC     1         -F           NO     DTE
```

Figure 5.1: Sample Link Status Display

2. To exit the link status display, enter **Exit:;** and press **Return**.

5.5.2 Daily Log Files

The system creates and stores daily log files in the **$XNV/log** directory. The system does not automatically delete older files. Older log files must be deleted to recover disk space.

5.5.3 Enabling and Disabling SS7 Message Logging

You can enable the SS7 message log file to allow capture of the SS7 messages that are received from the SS7 network and sent to Circuit Interworking, and the SS7 messages that are received from Circuit Interworking and sent to the SS7 network. This feature is useful for diagnosing problems and should not be enabled during normal operation.

To enable SS7 message logging, complete the following steps:

1. Log on as **cktint**.
2. Start the message logging process by entering the following command and pressing **Return**:

   ```
   ebslog on
   ```

   This command causes the messages to be logged in the file **/tmp/LOG.out.**

To disable SS7 message logging, complete the following steps:

1. Log on as **cktint**.
2. Kill the logd process by entering the following command and pressing **Return**:

   ```
   ebslog off
   ```

5.6 MANAGING CIRCUITS AND CIRCUIT GROUPS (ISUP_CONSOLE)

When Circuit Interworking is running, you can use the **isup_console** program, which provides you with the following management functions:

Blocking (b) — Generates a Block (BLO) or Circuit Group Block (CGB) message for the specified circuit or circuit group.

Debug (d) — Allows debug flags to be turned ON or OFF. Displays all debug flags that are turned ON. If no debug flags are ON, the message "No debug switches are active" is displayed.

NOTE: Use the "d" command to toggle each debug flag. The same debug functions can be specified in the CktInt.cfg file (refer to Section 4.5). Table 4.1 *lists and defines valid values for the debug flags.*

Feature Flags (f) — Allows feature flags to be enabled or disabled. Displays all the feature flags that have been enabled. If no feature flags are enabled, the message "No feature flags are enabled" is displayed.

NOTE: Use the "f" command to toggle each feature flag. The same feature flags can be specified in the CktInt.cfg file (refer to Section 4.5). Table 4.1 *lists and defines valid values for the feature flags.*

Help (h) — Lists all **isup_console** option definitions.

Ingest Circuit State (i) — Generates a request to the Circuit Interworking module to retrieve and ingest circuit state information from the ISUP layer of the SS7 stack.

Stop (k) — Stops a specified port from repeatedly sending a maintenance message (RSC, BLO, UBL, GRS, CGB or CGU) when it does not receive an appropriate network acknowledgment. Forces the circuit state to Active Idle.

Host Link Status (l) — Displays the status of the host computer TCP links. The values for Host Link Status are NOT CONFIGURED, OFFLINE, CONNECTED, and ONLINE_READY.

Load Sharing Mode (m) — Allows you to change the host load sharing mode. Load sharing modes are ROUNDROBIN or BROADCAST.

Reset (r) — Generates a Reset (RSC) or Group Reset (GRS) message for the specified circuit or circuit group.

Status Report (s) — Displays the current status of the requested circuit or circuit group, including the hardware state.

Switchover (S) — Causes a system switchover.

NOTE: There are two options available for switchover: 1) switchover with reset (initiates Autostart if it is configured) and 2) switchover without reset. The difference between the two options is that option 1 resets the previously active VCO/4K side after transfer.

Unblocking (u) — Generates an Unblock (UBL) or Circuit Group Unblock (CGU) message for the specified circuit or circuit group.

Exit (x) — Exits the **isup_console** program.

The **isup_console** program is located in the **~/sys/CktIntAnEnv** directory. This directory is aliased by **$XNV**.

To run **isup_console**, enter the following command and press **Return:**

```
isup_console
```

NOTE: CktInt must be running to use the isup_console program.

5.6.1 Examples

This section contains examples that describe how to use the **isup_console** program.

Example 1

To obtain either a Circuit or Group Status Report, complete the following steps:

1. Start isup_console. The main prompt (shown below) appears:

   ```
   Enter a command (b, d, f, h, i, k, l, m, r, s, S, u, x):
   ```

2. Type **s** and press **Return**. The following prompt requests input as to whether you want information on a circuit or circuit group.

   ```
   Enter a group or circuit (g, c):
   ```

3. Enter **c** for circuit. The following prompt appears:

   ```
   Enter a circuit ID:
   ```

4. Enter the circuit ID and press **Return**. The following prompt appears:

   ```
   Enter a trunk group:
   ```

5. Enter the trunk group number and press **Return**. A Circuit Status Report similar to the report in Figure 5.1 appears:

   ```
        SDS Port Address:    510
        SDS Port In Service: Yes
        SDS Card In Service: Yes
        SS7 Circuit Address: 1
        SS7 Circuit Type:    ISUP
        Circuit State:       Active
        SS7 Hdwr State:      Active
        Call State:          Idle
        Sub State:           Idle
        SDS Device:          0
        Circuit Name:        TGRP-1-CGRP-1-CIC-28
        Group Number:        1
        Inpulse Rule Number: 0
        Signalling Point:    0
        CIC (From Network):  1
        GCIC(From CktInt):   1
        Trunk Group:         1
        Resource Group:      20
        Control Host:        NONE

      Enter a command (b, d, f, h, i, k, l, m, r, s, S, u, x):
   ```

Figure 5.2: Sample Status Display

Example 2

To block a circuit, complete the following steps:

1. Start isup_console. The main prompt appears:

   ```
   Enter a command (b, d, f, h, i, k, l, m, r, s, S, u, x):
   ```

2. Type **b** and press **Return**. The following prompt requests input as to whether you want to block a circuit or circuit group:

   ```
   Enter a group or circuit (g, c):
   ```

3. Enter **c** for circuit. The following prompt appears:

   ```
   Enter a circuit ID:
   ```

4. Enter the circuit ID and press **Return**. The following prompt appears:

   ```
   Enter a trunk group:
   ```

5. Enter the trunk group number and press **Return**. The blocking message is generated for the circuit you specified, and the main prompt reappears.
6. To exit **isup_console**, type **x** and press **Return**.

Example 3

To display host link status, complete the following steps:

1. Start isup_console. The main prompt appears:

   ```
   Enter a command (b, d, f, h, i, k, l, m, r, s, S, u, x):
   ```

2. Type **l** and press **Return**. The following prompt appears:

   ```
   Enter the Link Identifier (1-8) or 'a' for all links
   ```

3. Enter the link identifier and press **Return**. The following messages appear:

   ```
   HOST                STATE                 SDS        STATE
   LINK0n:host-name    tcp_portlink_state    sds-name   tcp_portlink_state
   ```

   Where,

   n = the link number, **hostname** = the name of the host computer, and **tcp_port** = the hexadecimal value of the logical port address between the SS7 subsystem and the host system

   tcp port link state = NOT CONFIGURED, OFFLINE, CONNECTED, ONLINE_READY, ONLINE_RESTRICTED, or ONLINE_NOT_READY

   sds-name = the name of the VCO/4K and **tcp_port** = the hexadecimal value of the corresponding logical link between the SS7 subsystem and VCO/4K

Example 4

To switch the host load sharing mode, complete the following steps:

1. Start isup_console. The main prompt appears:

   ```
   Enter a command (b, d, f, h, i, k, l, m, r, s, S, u, x):
   ```

2. Type **m** and press **Return**. Prompts similar to the following appear:

   ```
   Current Host Load Sharing Mode = BROADCAST
   Toggle Host Load Sharing Mode = y/n
   ```

3. To switch the load sharing mode, type **y** and press **Return**. Prompts similar to the following appear:

   ```
   Current Host Load Sharing Mode = ROUNDROBIN
   Toggle Host Load Sharing Mode = y/n
   ```

4. Type **n** and press **Return**. The main prompt reappears.

5.7 INITIATING A SWITCHOVER

This subsection only applies to redundant configurations.

There are a number of ways to initiate a switchover. However, for optimum results, Cisco Systems recommends that you use only the following procedures, listed in order of preference:

1. Send a Change Active Controllers ($C0 01) command from the host. This command can be sent to either the active or the standby side of the VCO/4K. (Refer to the *Programming Reference* for information on the $C0 01 Command.)
2. Run the Switch Active Side to Standby utility from the VCO/4K System Administration Console on the active side of the system. (Refer to the *System Administrator's Guide* for information on the utility.)
3. Run the isup_console utility from the SS7 subsystem Administration Console on the active side of the system and issue an S command (refer to *Section 5.6*).

Switchover does not occur if the host is not connected to both sides, the VCO/4K standby side is not up and running, or file synchronization with the active side was unsuccessful.

> *NOTE: Switchovers initiated from the SS7 selector switch or MML commands are not supported. These methods do not send messages to the VCO/4K Generic and conflicts could arise, such as, the Generic attempting to switch from side A to side B, while the MML commands or the selector switch are attempting to switch from side B to side A.*

Do not initiate a switchover by flipping the A/B toggle switches on the VCO/4K Alarm Arbiter card or the SS7 selector switch. Keep both these A/B toggle switches in the AUTO position during normal operation, as shown in Figure 5.3 and Figure 5.4.

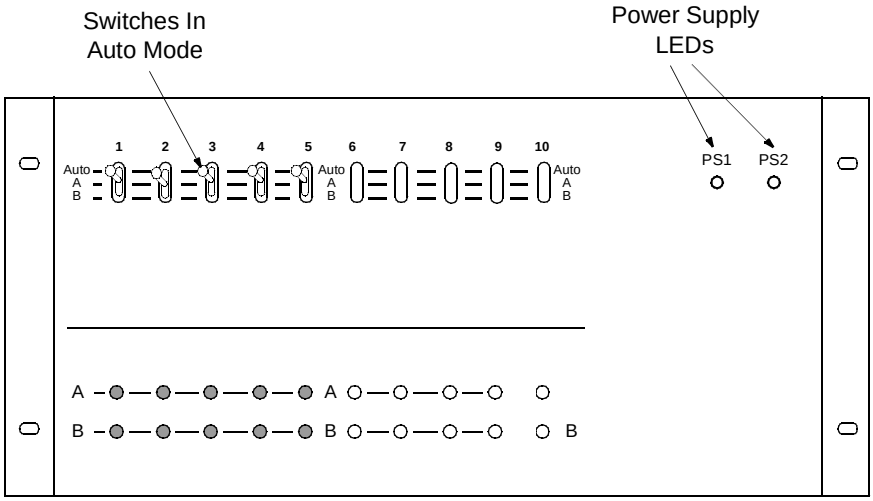


Figure 5.3: SS7 Selector Switch A/B Toggle Switch In AUTO Position

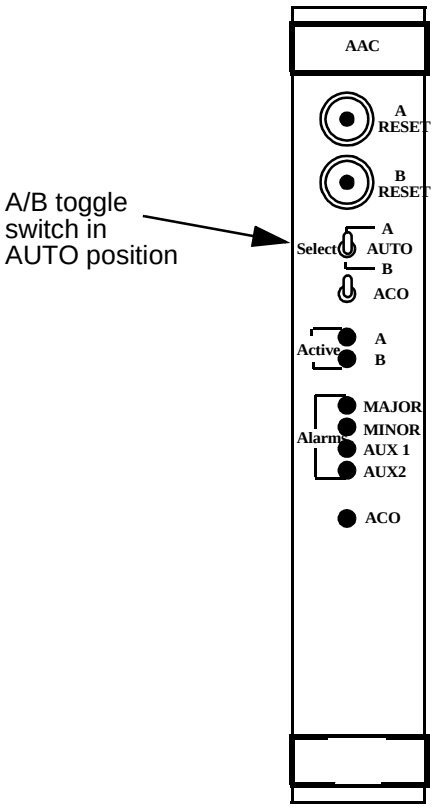


Figure 5.4: AAC A/B Toggle Switch In AUTO Position

When you perform maintenance, the A/B toggle switches can be used to lock on side A or side B after you initiate a switchover.

For example, if the SS7 subsystem and VCO/4K are active on side A, and you want to perform maintenance on side A, complete the following steps:

1. Send a Change Active Controllers ($C0 01) command from the host to the VCO/4K.
2. When the system has switched over to side B, set the toggle switches on the AAC and the SS7 selector switch to position B.
3. Perform the maintenance procedure on side A.
4. Set the toggle switches on the AAC and SS7 selector switch to AUTO. This forces the SS7 subsystem and VCO/4K to switch over so that the active side is side A.

If the SS7 subsystem and VCO/4K are on the side B, and you want to perform maintenance on side B, complete the following steps:

1. Send a Change Active Controllers ($C0 01) command from the host to the VCO/4K.
2. When the system has switched over to side A, set the toggle switches on the AAC and the SS7 selector switch to position A.
3. Perform the maintenance procedure on side B.
4. Set the toggle switches on the AAC and SS7 selector switch to AUTO. The SS7 subsystem and VCO/4K will remain active on side A.

5.8 BRINGING DOWN THE SS7 SUBSYSTEM

To bring down the SS7 subsystem, complete the following steps:

1. Log in as **sept**.
2. Stop SEPT by entering the following command and pressing **Return**:

   stop-sept
3. Switch user (**su**) to **cktint**.
4. Stop SS7 by entering the following command and pressing **Return**:

   stop-ss7.sh

   The following messages appear:

   ```
   This script will assist you in bringing down your Integrated
   SS7 system in a controlled fashion.

   Would you like to halt the Circuit Interworking (CktInt) software? [y/n]
   ```

5. Type **y** and press **Return**. Messages similar to the following appear:

```
Checking for process cktint...
Stopped cktint using kill -3 791
 792
.
Circuit Interworking is now stopped.

Would you like to halt the EBS software? [y/n]
```

6. Type **y** and press **Return**. Messages similar to the following appear:

```
Checking for process AccessISUP
Stopped AccessISUP using kill -9 638
Stopping EBSss7...

EBS SS7 stack is now stopped.
```

5.8.1 Powering Down the SPARC

To power down the SPARC, complete the following steps:

1. Log in as **root**.
2. Stop SS7 by entering following Step 4 through Step 6 above.
3. Type the following command a few times, pressing **Return** after each:

   sync
4. Shut down the SPARC by typing the following command and pressing **Return**:

   halt
5. Once you get a boot prompt (or wait one minute if you don't have a console hooked up), power down the SPARC by hitting the power button.

5.9 REMOTE ACCESS

The SS7 subsystem can be accessed from a remote location when the modem is connected to the system as described in *Section* 2.

To log in remotely, refer to the information in the *Solaris System & Network Administration* manual. Use the UNIX-to-UNIX copy commands (**ppp**) and Telnet/ftp commands. If you encounter any difficulties during remote login, contact Cisco Systems Technical Support.

5.10 USING SCRIPTS/ALIAS

The following alias and scripts help to streamline processes within Circuit Interworking. These are provided for convenience only; use the standard tools, such as isup_console or mml 0, if any problems arise.

NOTE: Cisco Systems does not recommend using these on a live production switch due to potentially unpredictable behavior.

Script/Alias Name	Function	Usage
dbs	Displays the CktInt debug flags that are turned ON.	dbs
dbc	Toggles the status of the debug flags specified. Type command name, followed by an empty space and the number of the flag(s) to be toggled (i.e. dbc 2 3 4).	dbc <debug flag(s)>
ebslog	Turns EBS logging ON/OFF.	ebslog <on/off>
msg	Displays the status of circuits associated with the input circuit group(s) as seen by EBS. Type command name, followed by a space, then the point code number and group number(s) (i.e. msg 1 1).	msg <pcno> <grpids>
msgmsp	Same as msg, but asks for additional SP numbers.	msgmsp <spno> <pcno> <grpids>
px	Prints information about SS7 processes now running with a full listing.	px
rcg	Issues a “reset circuit” for each circuit in the trunk group(s). Type command name, followed by space, and group number(s) (i.e. rcg 4).	rcg <trnkgrpid>
rg	Used to issue a “reset circuit group” for each input trunk group. Type command name, followed by space, and the range of number(s) (i.e. rg 1 -or- rg 1 2 3 -or- rg 1-3).	rg <trnkgrpid>
rmdb	An alias to remove EBS database files. Must be run for each SP.	rmdb <n>
sg	Displays the status of circuits associated with the input group(s) as seen by CktInt. Type command name, followed by space, and the group number(s) (i.e. sg 4). *NOTE: Output varies according to your specific system configuration.*	sg <trnkgrpid>
ss7-links	Displays status of SS7 links.	ss7-links
tcp-links	Displays current TCP link status.	tcp-links

Section 6
ISUP

6.1 INTRODUCTION

ISUP software consists of the Circuit Interworking software module and EBS SS7 stack.

The Circuit Interworking software module integrates SS7 signals with the VCO/4K, and host call processing and circuit maintenance tasks. It supports SS7-to-SS7 calls, SS7-to-VCO/4K calls, and VCO/4K-to-SS7 calls. Figure 6.1 is an illustration of Circuit Interworking.

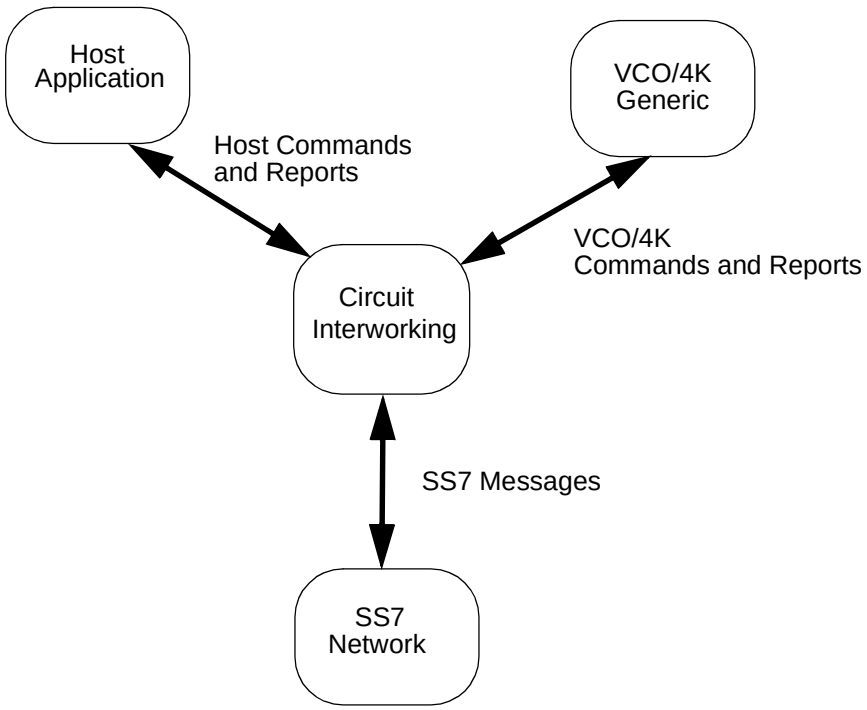


Figure 6.1: Circuit Interworking

The Circuit Interworking software communicates with the SS7 Network via SS7 message, and with the VCO/4K and host via commands and reports (see *Section 6.4*).

SS7 Messages

When the Circuit Interworking software receives an SS7 message, a host report is generated. Circuit Interworking may also generate an VCO/4K command and an SS7 message.

Refer to *Appendix E* for a list of the SS7 messages supported by circuit interworking.

6.2 TEMPLATES

The SS7 templates hide the parameter details of the SS7 message format, relieving host management. Templates also dictate the content of SS7 messages that are generated and transmitted to the SS7 network by Circuit Interworking in the $49 command (see *Section 6.5.3* or *Section 6.6.3*). The SS7 software supports templates 1 through 128. Circuit Interworking needs message templates 1 through 12, listed below, for its internal operations.

Template Number	SS7 ISUP Message Type
1	Initial Address Message (IAM)
2	Answer (ANM)
3	Address Complete (ACM)
4	Release (REL)
5	Release Complete (RLC)
6	Loopback Acknowledgment (LPA)
7	Continuity Check Request (CCR)
8	Call Progress (CPG)
9	Information (INF)
10	Information Request (INR)
11	Resume (RES)
12	Suspend (SUS)
13 to 128	User Specified

Templates specify some, none, or all of the SS7 message parameters. Template-specified parameters and host-specified parameters are combined to create SS7 messages to be transmitted to the SS7 network. A template has the following format:

[Message Type]

[Parameter Code] [Parameter Data Length] [Parameter Data]

[Parameter Code] [Parameter Data Length] [Parameter Data]

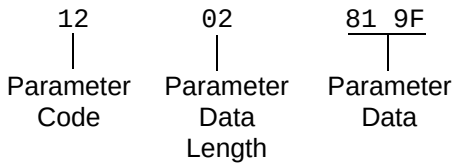
[Parameter Code] [Parameter Data Length] [Parameter Data]

The Message Type byte is the hex value of the message (refer to SS7 protocol for these values).

*NOTE: **Do not** modify message type data within SS7 templates 1 through 12.*

Use one line in the template file for each parameter in the SS7 message. You can use as many lines as necessary to specify the parameters.

Each parameter line consists of a Parameter Code byte, a Parameter Data Length byte, and one or more Parameter Data bytes. The format and components are defined as follows:



Parameter Code – Specifies the parameter code.

Parameter Data Length – Specifies the number of parameter data bytes; length of 00 indicates no parameter data.

Parameter Data – Specifies the parameter data exactly as it should be sent to the SS7 network. Code this data as it would appear in an SS7 message.

*NOTE: Templates furnished as starting points only. Template parameters **must be** reviewed and modified according to your specific application requirements to prevent potential operational problems with SS7 messages.*

Template files are numbered from 1 to 31 (decimal) and have the following naming convention: template_x, where *x* is the template number. When you enter a template number in a command, you must convert the decimal number to hexadecimal.

The template files are stored in the **$XNV/templates** directory. (**$XNV** is defined as **/export/home/cktint/sys/CktIntAnEnv**.) You can edit these template files with any standard UNIX text editor. Circuit Interworking must be restarted when changes are made to the template files.

6.3 CIRCUIT INTERWORKING STATE MACHINES

The Circuit Interworking software module contains a state machine for each of the various inputs from either the SS7 network, the host computer, or the VCO/4K.

Call control states include idle, outgoing busy, and incoming busy. Additionally, the Outgoing and Incoming Busy states include the following substates:

Outgoing Busy Substates	Incoming Busy Substates
Null	Null
Initiated	Call Present
Received	Released
Released	Delivered
Answered	Answered

NOTE: Additional substates can exist in certain call flow scenarios. For example, a continuity call flow scenario could produce Continuity, Continuity Initiated, or Continuity Call Present substates.

6.3.1 Basic State Machine

The basic state machine depicted in Figure 6.2 shows a list of events that cause transitions in states.

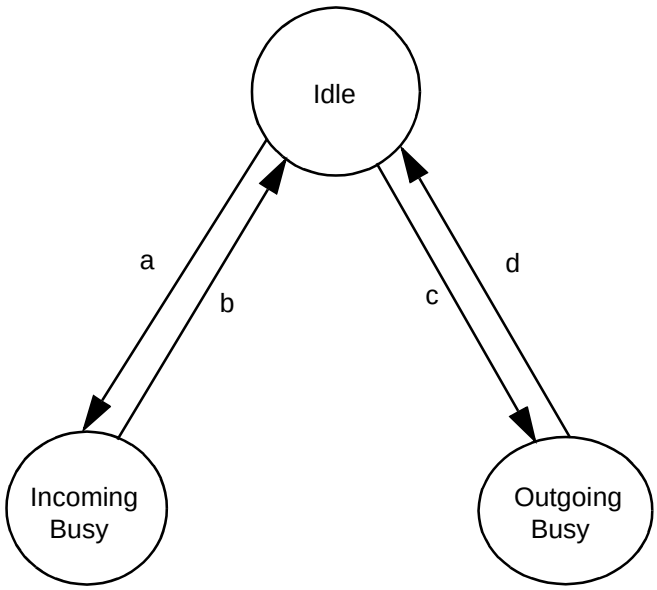


Figure 6.2: Basic State Machine

The transitions are defined as follows:

a. IAM message received from SS7 network. Essentially this is a basic call request. This is the state of call processing for a particular call.

b. RLC message sent to, or received from the SS7 network. This is the end of call processing for a particular call.

c. IAM message generated for an SS7 circuit. This is the start of call processing for an outgoing call.

d. RLC message sent of received to or from the SS7 network. This is the end of call processing for a particular call.

6.3.2 Outgoing Call Control States

The substates for the outgoing busy state are null, initiated, received, released, and answered (as shown in Figure 6.3).

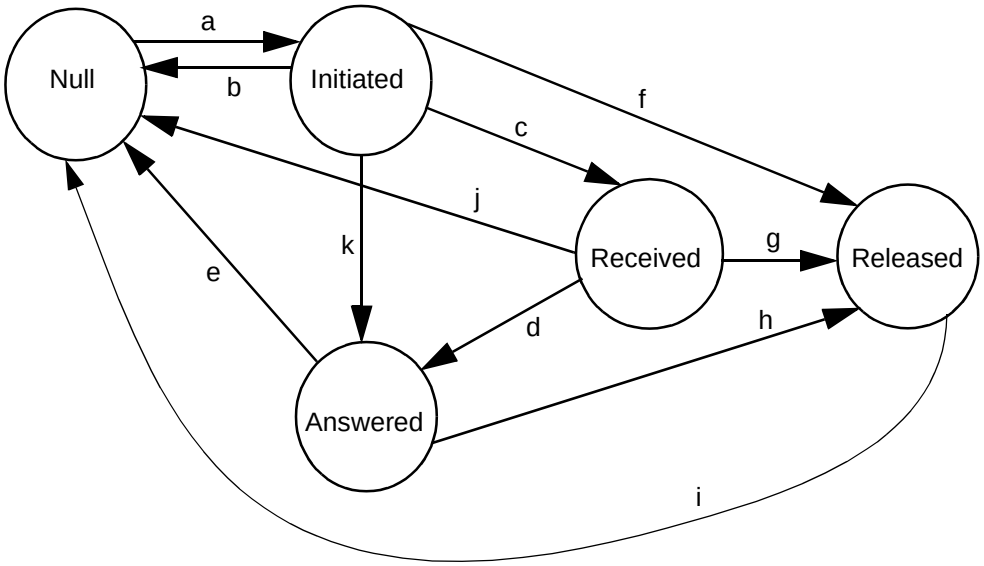


Figure 6.3: Substates for Outgoing Busy State

a. A successful IAM message sent to the SS7 network.
b. REL or RLC message received from the network. The REL or RLC is reported to the host as an $EA report with all the received parameters.
c. ACM message received from the network. The ACM is reported to the host as an $EA report with all the received parameters.
d. ANM message received from the network. The ANM is reported to the host as an $EA report with all the received parameters.
e. REL or RLC message received from the network. The REL or RLC is reported to the host as an $EA report with all the received parameters.
f. REL message sent to the network. Host command received requesting disconnect. Incoming SS7 circuit releases, release of the associated VCO/4K outgoing channel.
g. REL message sent to the network. Host command received requesting disconnect. Incoming SS7 circuit releases, release of the associated VCO/4K outgoing channel.
h. REL message sent to the network. Host command received requesting disconnect. Incoming SS7 circuit releases, release of the associated VCO/4K outgoing channel.
i. RLC message received from the network. The RLC is reported to the host as an $EA report with all the received parameters.
j. REL or RLC messages received from the network. The REL or RLC is reported to the host as an $EA report with all the received parameters.
k. CON message received from the network. The CON is reported to the host as an $EA report with all the received parameters.

6.3.3 Incoming Call Control States

A state diagram for the incoming call control states is shown in Figure 6.4. Following the figure is a description of each of the events that cause a transition from one state to the next.

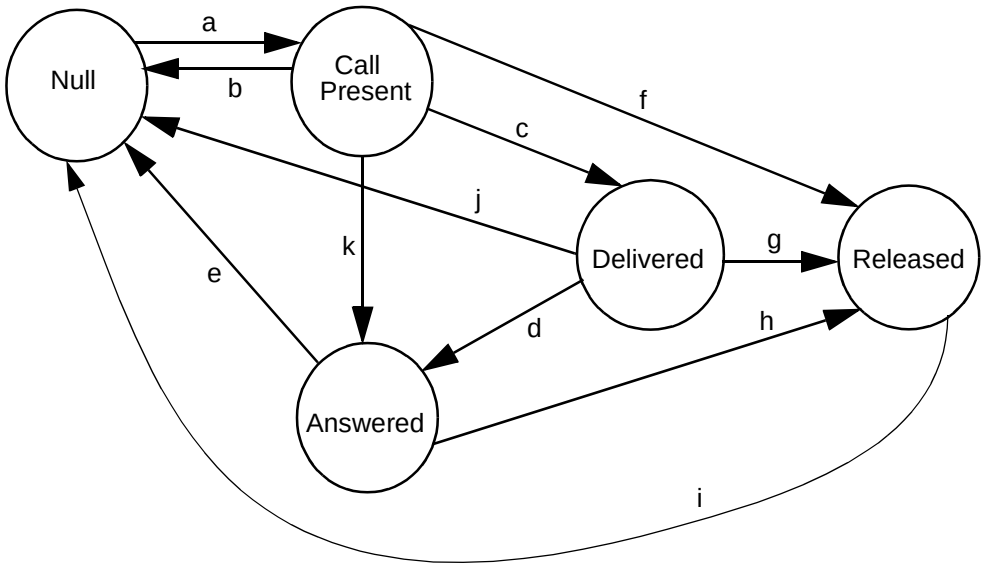


Figure 6.4: Incoming Call Control States

a. IAM messaged received. This is a call request. An $EA report to the host is generated, containing all of the received parameters.

b. REL or RLC message received from the network. The REL or RLC is reported to the host as an $EA report with all the received parameters.

c. ACM message generated to the SS7 network.

d. ANM message generated to the SS7 network.

e. REL or RLC received from the network. This event causes an $EA report to be generated, containing all received parameters.

f. REL generated to the network. Host command received to disconnect the incoming port.

g. REL generated to the network. Host command received to disconnect the incoming port.

h. REL generated to the network. Host command received to disconnect the incoming port.

i. RLC received from the network. This event causes an $EA report to be generated, containing all received parameters.

j. REL or RLC received from the network. Otherwise an $EA is generated, containing all received parameters.

k. ANM message received from the network. The CON is reported to the host as an $EA report with all the received parameters.

6.4 SS7 COMMANDS AND REPORTS OVERVIEW

The element that determines whether a command is *for* SS7 or the VCO/4K is the destination virtual communications address (VCA). Source VCAs determine whether a report is *from* SS7 or the VCO/4K. VCAs are part of the network header segment of all messages.

NOTE: Refer to the Message Structure Overview section in the Cisco VCO/4K Standard Programming Reference *or* Cisco VCO/4K Extended Programming Reference *for more information on VCAs.*

Commands

If the destination VCA in the message from the host is $C0, the command is an SS7 command and is processed by the SS7 subsystem. If the destination VCA in the message from the host is anything else, the SS7 subsystem passes the message on to the VCO/4K. Figure 6.5 illustrates how the SS7 subsystem handles SS7 commands.

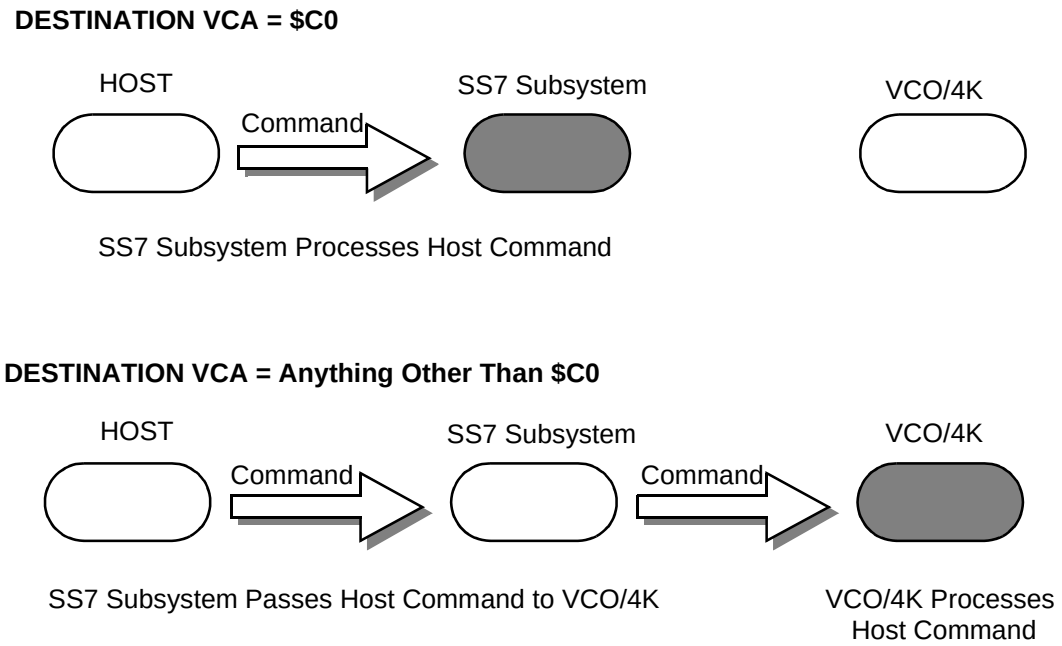


Figure 6.5: Handling of Commands by the SS7 Subsystem

Reports

If the source VCA in a message is $C0, the report is from SS7. If the source VCA in a message is anything else, the report is from the VCO/4K. Figure 6.6 illustrates how the SS7 subsystem handles SS7 reports.

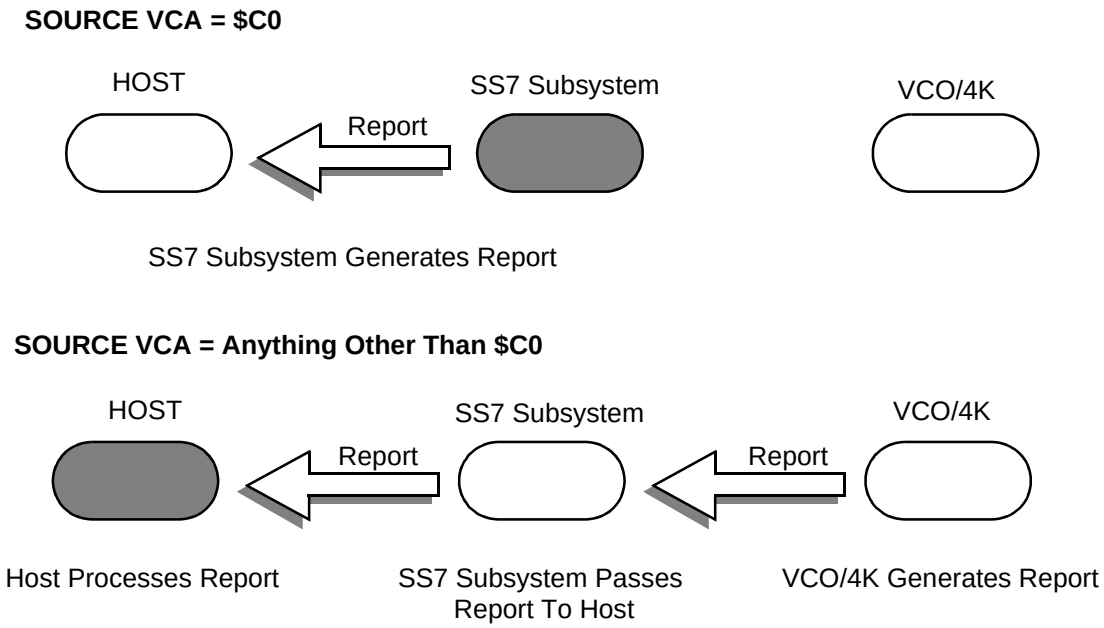


Figure 6.6: Handling of Reports by the SS7 Subsystem

Circuit Interworking processes the $30 01, $30 02, $49, $C0 04 and $C0 05 commands when the commands are addressed to the SS7 subsystem (i.e., the destination VCA is set to $CO). In addition, Circuit Interworking generates $B0 01, $D3, $D9, $EA, and $F0 reports in response to SS7 messages.

Table 6.1 and Table 6.2 list the commands and reports that are generated and processed by the Circuit Interworking software module.

Section 6.5 and *Section 6.6* describe SS7 commands and reports in greater detail. Refer to the *Cisco VCO/4K Standard Programming Reference* or *Cisco VCO/4K Extended Programming Reference* for additional information on non-SS7 commands and reports used by Circuit Interworking.

Table 6.1: Circuit Interworking Commands.

Command	Type	Description
$30 01	System Status	SS7 Circuit Query. Initiates query of SS7 circuit states for one or all trunk groups. Generated by the host and processed by cktint.
$30 02	Configuration Control	SS7 Circuit Synchronization. Synchronizes Circuit Interworking circuit states of one or all trunk groups. Generated by the host and processed by cktint.
$49[a]	Resource Control	SS7 Network Message Generation. Generates SS7 messages to the SS7 Network. Generated by the host and processed by cktint.
$C0 04[a]	Configuration Control	SS7 Call Processing Control. Used in conjunction with the Host Control of Call Load feature in the Generic software. When the feature is enabled, this command allows the host to indicate its ability to process all calls, existing calls only, or no calls. Generated by the host and processed by cktint.
$C0 05[a]	Configuration Control	SS7 Port Control. Allows a host process to relinquish control of a call assigned to itself, or assume control of a call that has been assigned to a different host. Generated by the host and processed by cktint.

a. Used in both SS7 (destination VCA of $CO) and non-SS7 (destination VCA not $CO) applications. For more information, refer to *Section 6.5* or *Section 6.6*.

Table 6.2: Circuit Interworking Reports

Report	Type	Description
$B0 01	Circuit Status	SS7 Circuit Status. Reports SS7 Circuit states for a trunk group. Generated by cktint for the host.
$D3[a]	System Status	SS7 System Port Status. Informs the host of a change in the status of an SS7 circuit. Generated by cktint for the host.
$D9[a]	Circuit Group Status	SS7 Circuit Group Status. Generated when cktint receives an incoming CGB/CGU/GRS from the network, system, or isup_console.
$EA[a]	Resource Control	SS7 Network Message Reception. Reports SS7 messages and parameters to the host. Generated by cktint for the host.
$F0[a]	System Status	SS7 System Alarm Status. Reports MTP OOS/IS link alarms. Generated by cktint for the host.

a. Can be either an SS7 (source VCA of $CO) or non-SS7 (source VCA not $CO) report. For more information, refer to *Section 6.5* or *Section 6.6*.

6.5 CIRCUIT INTERWORKING COMMANDS AND REPORTS (STANDARD)

This section describes the commands and reports used by the SS7 subsystem in Standard Mode. Refer to the *Programming Reference* for additional information on non-SS7 commands and reports used by Circuit Interworking.

6.5.1 SS7 Circuit Query ($30 01) Command (Standard)

Command Type: System Status

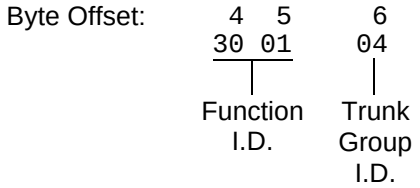
Description:

Initiates a query of the SS7 circuit states for one or all trunk groups. An SS7 Circuit Status ($B0 01) report is generated back to the host.

Usage Guidelines:

You may use this command at any time, but it is recommended that its use be restricted to periods of low traffic.

Format:



Function ID (byte offset 4 and 5) – Byte immediately following the Network Header. The Function ID uniquely identifies the command to the SS7 subsystem.

Trunk Group Id (byte offset 6) – Specifies the trunk group to query. Must be in the range from 0 to 127. Construct byte in binary according to the descriptions below, then convert to hex for use in the command.

XYYY YYYY

X = 0 — Query trunk group specified in YYY|YYYY

X = 1 — Query all trunk groups.

YYY|YYYY = 0 – 127 – Trunk group to query.

Network Status Byte:

The following network status bytes are returned by the $30 01 Command:

Status Byte	Meaning
$01	Command processing accepted. Indicates command sent to the system has passed initial processing. Returned only when Return All is specified in the command's Network Control byte. No corrective action required.
$D0	Invalid Trunk Group ID.
$41	Invalid host message length.

6.5.2 SS7 Circuit Sync ($30 02) Command (Standard)

Command Type: Configuration Control

Description:

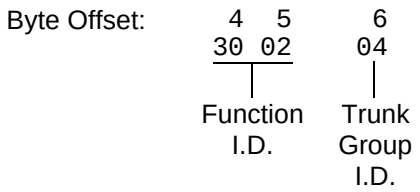
Synchronizes the CktInt circuit states of one or all trunk groups with the underlying ISUP layer.

Usage Guidelines:

This command may be used at anytime, but it is recommend that its use be restricted to periods of low traffic.

This command can be used in case the need arises to make sure the circuit states are consistent with what is maintained in the ISUP layer.

Format:



Function ID (byte offset 4 and 5) – Byte immediately following the Network Header. The Function ID uniquely identifies the command to the SS7 subsystem.

Trunk Group Id (byte offset 6) – Specifies the trunk group to sync. Must be in the range from 0 to 127. Construct byte in binary according to the descriptions below, then convert to hex for use in the command.

XYYY YYYY

X = 0 – Sync trunk group specified in YYY|YYYY

X = 1 – Sync all trunk groups.

YYY|YYYY = 0 – 127 – Trunk group to sync.

Network Status Byte:

The following network status bytes are returned by the $30 02 Command:

Status Byte	Meaning
$01	Command processing accepted. Indicates command sent to the system has passed initial processing. Returned only when Return All is specified in the command's Network Control byte. No corrective action required.
$D0	Invalid Trunk Group ID.
$41	Invalid host message length.

6.5.3 SS7 Network Message Generation ($49) Command (Standard)

Command Type: Resource Control

Description:

The SS7 $49 Command is used for all call control functions for an SS7 call, including connecting an incoming and outgoing port, and forcing SS7 call origination. Non-SS7 network interface ports can also be controlled with this command in an interworking scenario. The $49 Command functions as follows:

Set Disconnect Control – Each SS7 $49 Command for an SS7 controlled port sets the Disconnect Control values. Therefore the command must specify the desired Disconnect Control values. This allows it to be changed after a call has been established.

Create Call – To allocate an internal call object for maintaining connection relationships and take the SS7 controlled port(s) off-hook, you must set the Switching and Attaching bits in the Connection Control byte in the first SS7 $49 for the call.

Release Call – To deallocate the internal call object and put the SS7 controlled port(s) on-hook, you must set the Switching bit in the Connection Control byte of the $49 Command releasing the call. Do not set the Attaching bit.

Associate Ports – To make the system consider the Disconnect Control values in a $49 Command for a non-SS7 controlled port, associate the port with an SS7 controlled port in the call object. You can do this at anytime by providing the non-SS7 port as the other Port Address, either Controlling or Associated.

Send ISUP Message – Set the Template Control byte or the ISUP Message Type byte to send an ISUP message through the Associated Port Address.

- If both the Controlling and Associated Port Addresses are not SS7 controlled or not specified, then network status byte $03 is returned.
- If both the Template Control byte and the ISUP Message Type byte are zero (0), then no ISUP message is sent.

Usage Guidelines:

To use the SS7 $49 Command to generate an ISUP message, the destination VCA must be set to $C0, and either byte offset 7 (ISUP Message Type) must be set for the message type, or byte offset 19 (Template Control) must be set to the template number.

Use this command for SS7 call control functions, including connecting an incoming and outgoing port, and forcing SS7 call origination. Control over the contents of SS7-generated messages is provided.

Total command length cannot exceed 306 bytes.

The controlling port is specified by port address.

A virtual port is not required to originate an outgoing call. Set the Control Address ID to 0, if you want the Controlling Port Address to be used by the host for tracking purposes.

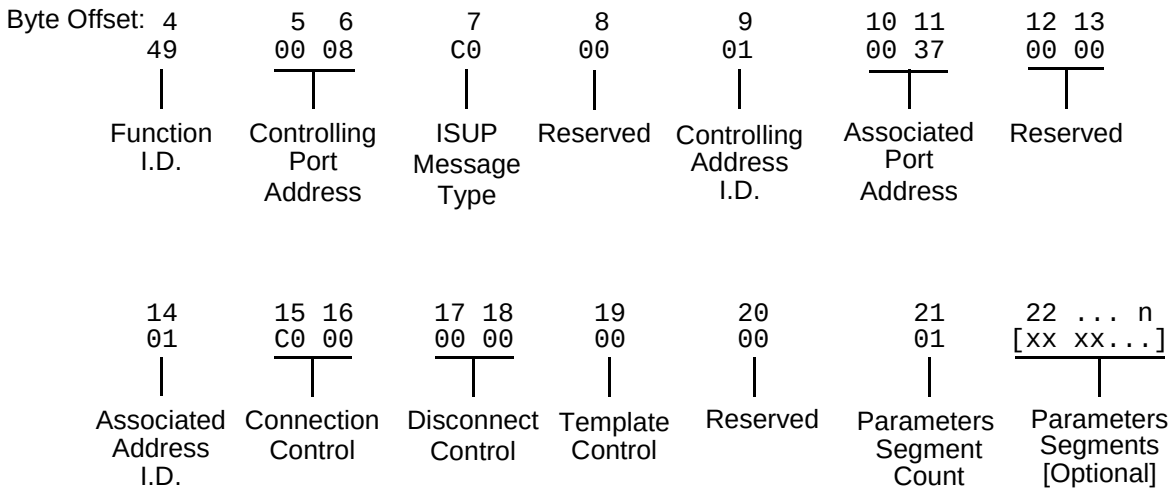
You can choose the associated (outgoing) port two ways: specify by port address or hunt through a resource group.

The returned command is truncated to report only through byte offset 14 (Associated Address I.D.).

In interworking scenarios, either the controlling or associated port can be a non-SS7 port.

If you specify parameters, their values must be coded as they would appear in an SS7 message. Parameters you specify in the command are included in the next SS7 message transmission. The host-specified parameters are added to any SS7 parameters included in the template being executed.

Format:



Function ID (byte offset 4) – Byte immediately following the Network Header; uniquely identifies the command to Circuit Interworking.

Controlling Port Address (byte offset 5 and 6) – Hex representation of the controlling port circuit address for which the command is sent.

ISUP Message Type (byte offset 7) – The ISUP messages to generate only if the Template Control byte (byte offset 19) is set to zero.

Reserved (byte offset 8 – Always 00

Controlling Address Identifier (byte offset 9) – Specify the byte according to the following:

00 – no controlling address used in command. Controlling Port Address may be used by the host for tracking purposes. (To initiate outbound SS7 calls, set to 00.)

01 – controlling port specified by port address

Associated Port Address (byte offsets 10 and 11) – Hex representation of the associated (outgoing) port circuit address for which the command is sent or the resource group number from which to select the outgoing port. The Associated Address Identifier (byte offset 14) must be set to $01.

Reserved (byte offsets 12 and 13) – These bytes must be set to $00 00.

Associated Address Identifier (byte offset 14) – Specifies that the associated (outgoing) port is identified by port address/resource group. Specify the byte according to the list below:

00 – no associated address used in command

01 – associated port specified by port address, or resource group number

Connection Control (byte offsets 15 and 16) – Specifies the switching, attaching, and hunting options when this command is used to connect a call. Byte offset 16 is reserved for future enhancements and must be set to $00. Construct byte offset 15 according to the descriptions below, then convert to hex for use in the command.

SAPVV000

S – Specifies if switching action is required

S = 0 – no switching action required; A bit ignored and should set to 0

S = 1 – switching action required

A—Specifies whether to link or remove a resource

A = 0 – if S = 1, remove resource from call; if S = 0, no meaning

A = 1 – if S = 1, link resource to call; if S = 0, no meaning

P – Specifies whether to use a specific outgoing circuit or to select any outgoing circuit from a resource group; port address or group number is specified in Associated Port Address bytes

P = 0 – for S = 0 or 1, use port specified in Associated Port Address bytes.

P = 1 – for S = 1, select port from resource group specified in Associated Port Address bytes; the port address of the selected channel is specified in the command returned to the host

VV – Not used for SS7. Set to 00.

Disconnect Control Bytes

The Disconnect Control bytes are always included in the command. You can only define byte offset 17 at this time. Byte offset 17 specifies the disposition of ports when the call is torn down. Byte offset 18 is reserved for future enhancements and should be set to $00.

Bit settings in this byte are overridden if another $49 Command is processed for either port in this $49 Command.

Disconnect Control (byte offset 17 & 18) – Determines what actions to take on a port when the opposite end goes on hook. Construct byte in binary according to the descriptions below, then convert to hex for use in the command.

0000I0U0

I – Specifies whether to return the incoming port to CP_SETUP state when the outgoing port goes on hook

I = 0 – force incoming to idle

I = 1 – set incoming to setup state upon outgoing disconnect; T must be 0

U – Specifies whether to return the outgoing port to CP_SETUP state when the in coming port goes on hook

U = 0 – force outgoing to idle

U = 1 – set outgoing to setup state upon incoming disconnect; C must be 0

Template Control Byte

The Template Control byte is always included in the command, and specifies the template to be used when transmitting an SS7 message.

Template Control (byte offset 19) – Specifies whether a template is to be used in this command and the template number, if any (see *Section 6.2*). Construct byte in binary according to the descriptions that follow, then convert to hex for use in the command.

X0RRRRRR

X – Specifies if an template is used in this command

X = 0 – no template specified. Use ISUP Message Type byte (offset byte 7).

X = 1 – execute template specified in RRRRRR; I must be 0

RRRRRR – Specifies the template number.

Reserved (byte offset 20) – Always 00.

Parameter Segment Count Byte

The Parameter Segment Count byte is used in all $49 Commands and specifies how many parameters are included in the command.

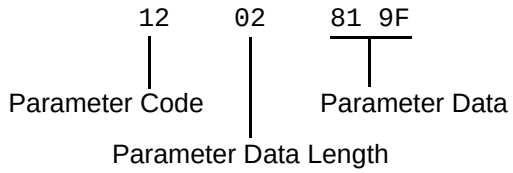
Parameter Segment Count (byte offset 21) – Specifies how many parameters are included in the command. Convert from decimal to hex for use in the command. A value of $00 indicates that no parameters follow this byte.

Parameter Segment

The Parameter Segment bytes contain parameters to be transmitted to the SS7 network. The number of parameters contained in this segment appears in the Parameter Segment Count byte.

> *NOTE: Each time a $49 command is sent, the parameters defined in that particular command override any permanent parameters set in the template it names in byte offset 19.*

Each parameter consists of a Parameter Code byte, a Parameter Data Length byte, and one or more Parameter Data bytes. The format and components of the Parameter Segment are defined as follows:



Parameter Code – Specifies the parameter code.

> *NOTE: When using the AOC feature, this must be specified as APP (Application Transport Parameter—0x78). See Section 6.7 for additional information on special coding for this parameter.*

Parameter Data Length – Specifies the number of parameter data bytes.

Parameter Data – Specifies the parameter data exactly as it should be sent to the SS7 network.

Coding for Pass Along Message (PAM)

End-to-End signalling uses the pass along method of signalling in which information is exchanged between two end points via the Pass Along Message (PAM). The contents of this message are relevant only to the end points; all other intermediate exchanges just pass it along transparently.

PAM is characterized by a special message type code as specified in ITU Recommendation Q.763. It has a value of 0x28. One PAM message may have one embedded ISUP message to be passed along.

Byte offset 7 of the $49 command must be set to 0x28 when a PAM message is sent. Then, the coding from byte offset 21 onward is shown in Figure 6.7.

This example shows the parameter segment count and parameter segment for a PAM with a Call Progress Message (CPG) as the embedded ISUP message.

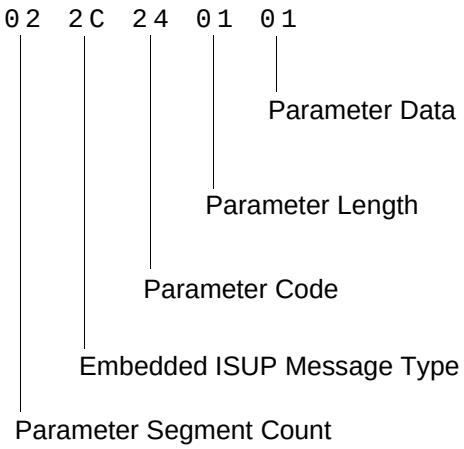


Figure 6.7: Example of CPG Message Embedded in a PAM

Parameter Segment Count – Specifies the number of parameters in the embedded ISUP message + 1 [the extra count is for the message type of the embedded ISUP message].

Embedded ISUP Message Type – Specifies the message type of the embedded ISUP message; no associated length and data information will follow.

Parameter Code – Specifies the parameter code of the parameter in the embedded ISUP message.

Parameter Length – Specifies the number of parameter data bytes.

Parameter Data – Specifies the parameter data exactly as it should be sent to the SS7 network.

Network Status Byte:

The following network status bytes are returned by the $49 Command:

Status Byte	Meaning
$01	Command processing successful. Indicates command sent to the system has been processed successfully. Returned only when Return All is specified in the command's Network Control byte. No corrective action required.
$03	Syntax error in command. Indicates one or more of the values specified in the command are invalid or that the command specifies no action or improper switching actions within the SAP bit settings. Specifying values in spacer bytes or reserved bytes may also cause syntax errors. Check all byte values in command.
$08	Command received was received by Standby side but can only be processed on the Active side.
$0D	Invalid resource group number. The resource group specified in the command is 0 or greater than 63. Check resource group number.
$0E	Invalid controlling port address (not in valid range). The port address specified in the command is not within the range of system port addresses or that port address is not assigned. Check port address.
$0F	Invalid controlling host
$15	Invalid associated port address (not in valid range). Indicates the port address specified in the command is not within the range of system port addresses. Check port address or select a port from appropriate resource group.
$2B	The Template number specified in the command is out of the range for a rule (1 to 32). Check value and resend command with correct value.
$41	Invalid host message length.
$C7	Invalid range specified in group maintenance messages.
$C8	Incoming port is in the wrong state.
$C9	No resource available.
$CA	Outgoing port is in the wrong state.
$CB	Parameter error in the host command.
$CC	Parameter error in the template file.
$CD	Invalid ISUP message type.
$CE	Missing Range and Status parameter.
$CF	Protocol violation. This network status byte usually indicates that the ISUP message type is invalid for the current call state.

Status Byte	Meaning
$D1	The Far End SP is unavailable.
SD4	Circuit in transient state. When a circuit is in maintenance transient state, no messages are allowed on that circuit.
$D5 (AOC only)	Invalid SLR (Segmentation Local Reference). Indicates the host application sent charging information segmented into two or more $49 commands and there was a mismatch in the outgoing SLR between the segments. Thus, the host command(s) were rejected by cktint.
$D6 (AOC only)	Invalid host cktint SI (Segment Indicator). Indicates the host application sent charging information segmented into two or more $49 commands and there was a mismatch in the outgoing SI between the segments. Thus, the host command(s) were rejected by cktint.

6.5.4 SS7 Host Call Load Control ($C0 04) Command (Standard)

Command Type: Configuration Control

Description:

Used in conjunction with the Host Control of Call Load feature in the Generic software (refer to the System Host Configuration screen in the *System Administrator's Guide*). When the feature is enabled, this command allows the host to indicate its ability to process all calls, existing calls only, or no calls.

Usage Guidelines:

The system verifies that the Host Control of Call Load feature has been enabled before processing the command. If the feature is not enabled in the Generic software, the command is returned with a network status byte of $1B (feature not enabled) in the network header.

If the load control code is set to 00, normal call processing occurs with no restrictions.

If the load control code is set to 01, cktint does not send SS7 $EA reports to the host or process SS7 $49 commands containing call control messages from the host until the host sends an SS7 $C0 04 command indicating that it is ready to process calls. All maintenance reports are still sent to the host; all maintenance message commands are also sent and processed.

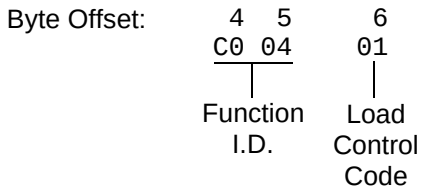
In addition, if the load control code is set to 02, cktint does not send new call reports to the host or process commands containing IAM messages from the host until the host indicates that it is ready to process new calls. However, commands will still be accepted from and reports sent to the host for calls in progress.

You can use the SS7 $C0 04 command various ways. For example, sending an SS7 $C0 04 command with the load control code set to 01 would allow the host to perform some type of initialization before accepting calls (the default setting on startup when the Host Control of Call Load feature is enabled). With the load control code set to 02, current calls will be completed, but no new calls accepted—a "graceful shutdown."

This command only affects operations with respect to the host from which it is received. If there are multiple hosts, the command must be sent by each host.

> *NOTE:* ***On an SS7 system, the $C0 04 command destination VCA must be set to $C0****; otherwise, this could result in tcp link state mismatches between the VCO/4K, cktint, and the host.*

Format:



Function ID (byte offset 4 and 5) – Byte immediately following the Network Header. The Function ID uniquely identifies the command to the SS7 subsystem.

Load Control Code (byte offset 6) – Determines whether the system should process incoming calls by reporting them to the host or reject call control commands sent by the host. Specify this byte according to the following list:

00 = Normal call processing

01 = Call processing suspended

02 = No new calls accepted or sent from the host; process existing calls only

Network Status Byte:

The following network status bytes are returned by the SS7 $C0 04 Command:

Status Byte	Meaning
$01	Command processing accepted. Indicates command sent to the system has passed initial processing. Returned only when Return All is specified in the command's Network Control byte. No corrective action required.
$03	Syntax error in command.
$1B	Host Control of Call Load feature is not enabled and command was not processed.
$D2	VCO/4K is not connected and command was not processed.

Host Link States:

The following table summarizes possible Host Link States associated with the SS7 $C0 04 Command:

Host Link State	Affect on Messaging
ONLINE_READY	Normal messaging. All host commands and reports allowed. Set by SS7 $C0 04 host command with the load control code set to 00.
ONLINE_NOT_READY	Call processing effectively suspended; all call control commands rejected. • No new SS7 $EA call reports sent. • No call control commands allowed; only configuration commands (i.e., $C0 04, $C0 01, $30 01, $30 02) and maintenance messages. Set by SS7 $C0 04 host command with the load control code set to 01.
ONLINE_RESTRICTED	Processing of calls in progress; host commands accepted. • No SS7 $EA report for new call reports for incoming IAM. • All host commands allowed, except SS7 $49 command with IAM. Set by SS7 $C0 04 host command with the load control code set to 02.

NOTE: If calls are present when the host goes ONLINE_NOT_READY, there can be a mismatch of circuit states among the VCO/4K, cktint, and the host.

When the Host Control of Call Load feature is enabled and there is a connection to the host, the initial tcp link state in cktint is ONLINE_NOT_READY. *It stays in this state until receiving an SS7 $C0 04 command from the host requesting a host link state change.*

6.5.5 SS7 Host Assume/Relinquish Port Control ($C0 05) Command (Standard)

Command Type: Configuration Control

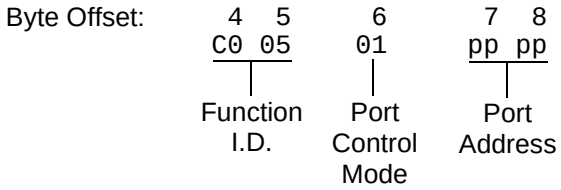
Description:

Allows a host process to relinquish control of a call assigned to itself, or assume control of a call that has been assigned to a different host.

Usage Guidelines:

This command performs the assume/relinquish port control function. A function code allows a host link to override a controlling host assignment with one of two operating modes: one mode allows a host to assume control of a port; the other mode allows a host to relinquish control of a port. In either case, the assignment override remains in effect as long as the affected port is involved in a call, or until a subsequent override command is received. The relinquish mode clears the controlling host assignment for the specified port so that another host can take control of that port.

Format:



Function ID (byte offset 4 and 5) – Byte immediately following the Network Header. The Function ID uniquely identifies the command to the SS7 subsystem.

Port Control Mode (byte offset 6) – Determines the type of action to be taken. Specify this byte according to the following list:
00 = Relinquish control
01 = Assume control

Port Address (byte offsets 7 and 8) – Port address for which control is being seized or relinquished.

Network Status Byte:

The following network status bytes are returned by the $C0 05 Command:

Status Byte	Meaning
$01	Command processing accepted. Indicates command sent to the system has passed initial processing. Returned only when Return All is specified in the command's Network Control byte. No corrective action required.
$03	Syntax error in command. Indicates one or more of the values specified in the command are invalid or that the command specifies no action or improper switching actions within the SAP bit settings. Specifying values in spacer bytes or reserved bytes may also cause syntax errors. Check all byte values in command.
$0E	Invalid controlling port address (not in valid range). Indicates the port address specified in the command is not within the range of system port addresses or that port address is not assigned. Check port address.
$C9	No resource available.

6.5.6 SS7 Circuit Status ($B0 01) Report (Standard)

Report Type: System Status

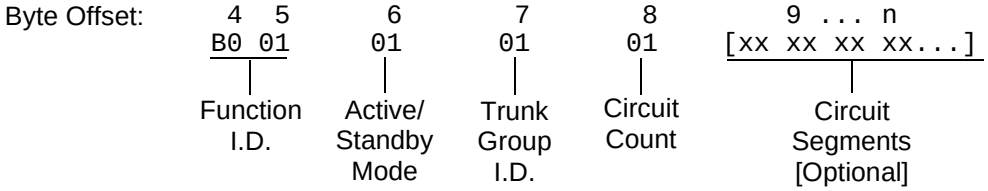
Description:

Reports the SS7 circuit states for a trunk group.

Action Causing Report Generation:

Generated in direct response to an SS7 Circuit Query ($30 01) Command. If all trunk groups were requested, then multiple reports are generated, one for each trunk group.

Format:



Main Command Segment

Function ID (byte offsets 4 and 5) – Bytes immediately following the Network Header; uniquely identifies this report from Circuit Interworking.

Active/Standby Mode (byte offset 6) – Indicates active/standby mode. Non-redundant systems will always return active mode.

0 = Standby mode

1 = Active mode

Trunk Group ID (byte offset 7) – Specifies the trunk group.

Circuit Count (byte offset 8) – Indicates the number of circuit segments that follow. This will be a value from 0 to 32.

Circuit Segments

Circuit Segments (byte offset 9 – n) – Each segment consists of four bytes. They are the Circuit Port Address (2 bytes), the Circuit State/Call State (1 byte), and the Circuit Sub State (1 byte). The number of circuit segments is specified by the Circuit Count.

Circuit Port Address – The VCO/4K port address associated with this circuit.

Circuit State/Call State — The (maintenance) state of this circuit. The possible hex values are as follows:

FF	Unknown
00	Unequipped
01	Transient
10	Unequipped/Idle
12	Active/Idle
13	Remotely Blocked/Idle
14	Locally Blocked/Idle
15	Locally and Remotely Blocked/Idle
16	Locally Block_Transient/Idle
17	Locally Unblock_Transient/Idle
18	Locally Block_Transient and Remotely Blocked /Idle
19	Locally Unblock_Transient and Remotely Blocked /Idle
22	Active/Incoming Busy
23	Remotely Blocked/Incoming Busy
24	Locally Blocked/Incoming Busy
25	Locally and Remotely Blocked/Incoming Busy
26	Locally Block_Transient/Incoming Busy
27	Locally Unblock_Transient/Incoming Busy
28	Locally Block_Transient and Remotely Blocked/Incoming Busy
29	Locally Unblock_Transient and Remotely Blocked/Incoming Busy
32	Active/Outgoing Busy
33	Remotely Blocked/Outgoing Busy
34	Locally Blocked/Outgoing Busy
35	Locally and Remotely Blocked/Outgoing Busy
36	Locally Block_Transient/Outgoing Busy
37	Locally Unblock_Transient/Outgoing Busy
38	Locally Block_Transient and Remotely Blocked/Outgoing Busy
39	Locally Unblock_Transient and Remotely Blocked/Outgoing Busy
82	Hardware Remotely Blocked, Maintenance Active/Idle
83	Hardware Remotely Blocked, Maintenance Remotely Blocked/Idle
84	Hardware Remotely Blocked, Maintenance Locally Blocked/Idle

85 Hardware Remotely Blocked, Maintenance Locally and Remotely Blocked/Idle

86 Hardware Remotely Blocked, Maintenance Locally Block_Transient/Idle

87 Hardware Remotely Blocked, Maintenance Locally Unblock_Transient/Idle

88 Hardware Remotely Blocked, Maintenance Locally Block_Transient and Remotely Blocked/Idle

89 Hardware Remotely Blocked, Maintenance Locally Unblock_Transient and Remotely Blocked/Idle

92 Hardware Locally Blocked, Maintenance Active/Idle

93 Hardware Locally Blocked, Maintenance Remotely Blocked/Idle

94 Hardware Locally Blocked, Maintenance Locally Blocked/Idle

95 Hardware Locally Blocked, Maintenance Locally and Remotely Blocked/Idle

96 Hardware Locally Blocked, Maintenance Locally Block_Transient/Idle

97 Hardware Locally Blocked, Maintenance Locally Unblock_Transient/Idle

98 Hardware Locally Blocked, Maintenance Locally Block_Transient and Remotely Blocked/Idle

99 Hardware Locally Blocked, Maintenance Locally Unblock_Transient and Remotely Blocked/Idle

A2 Hardware Locally and Remotely Blocked, Maintenance Active/Idle

A3 Hardware Locally and Remotely Blocked, Maintenance Remotely Blocked/Idle

A4 Hardware Locally and Remotely Blocked, Maintenance Locally Blocked/Idle

A5 Hardware Locally and Remotely Blocked, Maintenance Locally and Remotely Blocked/Idle

A6 Hardware Locally and Remotely Blocked, Maintenance Locally Block_Transient/Idle

A7 Hardware Locally and Remotely Blocked, Maintenance Locally Unblock_Transient/Idle

A8 Hardware Locally and Remotely Blocked, Maintenance Locally Block_Transient and Remotely Blocked/Idle

A9 Hardware Locally and Remotely Blocked, Maintenance Locally Unblock_Transient and Remotely Blocked/Idle

B2 Hardware Locally Block_Transient, Maintenance Active/Idle

B3 Hardware Locally Block_Transient, Maintenance Remotely Blocked/Idle

B4 Hardware Locally Block_Transient, Maintenance Locally Blocked/Idle

B5 Hardware Locally Block_Transient, Maintenance Locally and Remotely Blocked/Idle

B6 Hardware Locally Block_Transient, Maintenance Locally Block_Transient/Idle

B7 Hardware Locally Block_Transient, Maintenance Locally Unblock_Transient/Idle

B8 Hardware Locally Block_Transient, Maintenance Locally Block_Transient and Remotely Blocked/Idle

B9 Hardware Locally Block_Transient, Maintenance Locally Unblock_Transient and Remotely Blocked/Idle

C2 Hardware Locally Unblock_Transient, Maintenance Active/Idle

C3 Hardware Locally Unblock_Transient, Maintenance Remotely Blocked/Idle

C4 Hardware Locally Unblock_Transient, Maintenance Locally Blocked/Idle

C5 Hardware Locally Unblock_Transient, Maintenance Locally and Remotely Blocked/Idle

C6 Hardware Locally Unblock_Transient, Maintenance Locally Block_Transient/Idle

C7 Hardware Locally Unblock_Transient, Maintenance Locally Unblock_Transient/Idle

C8 Hardware Locally Unblock_Transient, Maintenance Locally Block_Transient and Remotely Blocked/Idle

C9 Hardware Locally Unblock_Transient, Maintenance Locally Unblock_Transient and Remotely Blocked/Idle

D2 Hardware Locally Block_Transient and Remotely Blocked, Maintenance Active/Idle

D3 Hardware Locally Block_Transient and Remotely Blocked, Maintenance Remotely Blocked/Idle

D4 Hardware Locally Block_Transient and Remotely Blocked, Maintenance Locally Blocked/Idle

D5 Hardware Locally Block_Transient and Remotely Blocked, Maintenance Locally and Remotely Blocked/Idle

D6 Hardware Locally Block_Transient and Remotely Blocked, Maintenance Locally Block_Transient/Idle

D7 Hardware Locally Block_Transient and Remotely Blocked, Maintenance Locally Unblock_Transient/Idle

D8 Hardware Locally Block_Transient and Remotely Blocked, Maintenance Locally Block_Transient and Remotely Blocked/Idle

D9 Hardware Locally Block_Transient and Remotely Blocked, Maintenance Locally Unblock_Transient and Remotely Blocked/Idle

E2 Hardware Locally Unblock_Transient and Remotely Blocked, Maintenance Active/Idle

E3 Hardware Locally Unblock_Transient and Remotely Blocked, Maintenance Remotely Blocked/Idle

E4 Hardware Locally Unblock_Transient and Remotely Blocked, Maintenance Locally Blocked/Idle

E5 Hardware Locally Unblock_Transient and Remotely Blocked, Maintenance Locally and Remotely Blocked/Idle

E6 Hardware Locally Unblock_Transient and Remotely Blocked, Maintenance Locally Block_Transient/Idle

E7 Hardware Locally Unblock_Transient and Remotely Blocked, Maintenance Locally Unblock_Transient/Idle

E8 Hardware Locally Unblock_Transient, and Remotely Blocked Maintenance Locally Block_Transient and Remotely Blocked/Idle

E9 Hardware Locally Unblock_Transient and Remotely Blocked, Maintenance Locally Unblock_translock_Transient and Remotely Blocked/Idle

Circuit Sub State — The call state of this circuit. The possible hex values are as follows:

00 Idle

01 Delivered

02 Present

03 Answered

04 Released

05 Received

06 Initiated

07 Continuity

08 Reserved

09 Continuity Reserved

0A Continuity Present

0B Continuity Initiated

0C Continuity Initiated Detected

0D Continuity Initiated End

0E Continuity End

0F Continuity Previous Call Present

10 Continuity Previously Initiated

11 Locally Suspended

12 Remotely Suspended

13 Local and Remote Suspended

6.5.7 SS7 System Port Status ($D3) Report (Standard)

Report Type: System Status

Description:

Informs the host of an attempted change in the status of a system resource port. Change can be the result of:

- Activating/deactivating a port or card span (refer to the *System Administrator's Guide*)
- Transmission or reception of an SS7 ISUP Maintenance message

Action Causing Report Generation:

This report is generated when a change occurs in the operating status of an individual SS7 circuit. Status changes can be caused by an action at the system administrative console (isup_console), a host command, or a received SS7 ISUP maintenance message.

A report indicating a port's availability may be returned from a circuit in Active mode. When a circuit is Out-of-Service, no other $D3 reports are generated for that port.

Format:

Byte Offset:	4	5	6 7	8	9	10 11 12
	D3	13	00 18	02	03	00 00 00
	Function I.D.	Old Status/ New Status	Port Address	Originator	Resource Group	Reserved

Function I.D. (byte offset 4) – Byte immediately following the Network Header; uniquely identifies the report from the system.

Old Status/New Status (byte offset 5) – Specifies the status of the port before the change occurred and the present status of the port. Convert byte from hex to binary and interpret the bits as described on the following page.

MMMMNNNN

MMMM – specifies the status of the port before the change occurred

MMMM = 0001 – resource was unavailable (Out of Service)

MMMM = 0011 – resource was on line and available

NNNN – specifies the current status of the port

NNNN = 0001 – resource is currently unavailable (Out of Service)

NNNN = 0011 – resource currently on line and available

Port Address (byte offsets 6 and 7) – Hex representation of port address for which the report is generated.

Originator (byte offset 8) – Specifies whether the change in status was originated by the system or the host and the reason for the change. The following originator code values are added to the CktInt $D3 reports sent to the host during SS7 events:

0x01 – A circuit group block (CGB) or a block (BLO) message was received from the SS7 network (SS7_BLOCKING event).

0x02 – A circuit group unblock (CGU) or a unblock (UBL) message was received from the SS7 network (SS7_UNBLOCKING event).

0x03 – A circuit group unblock acknowledge (CGUA) or an unblock acknowledge (UBA) message was received from the SS7 network (SS7_UNBLOCKING_ACK event).

0x04 – A reset circuit (RSC), circuit group reset (GRS), release complete (RLC), or circuit group reset acknowledge (GRA) message was received from the SS7 network (SS7_RESET event).

0x05 – A change of port status from OFFLINE to ONLINE was received from the VCO/4K in the form of $D9 or $D3 reports (SDS_UNBLOCK event).

0x06 – A change of port status to OFFLINE was received from the VCO/4K in the form of $D9 or $D3 reports (SDS_BLOCK event).

0x07 – The user initiated a circuit group unblock (CGU) or unblock (UBL) message either through isup_console or through a host message (MAINT_UNBLOCK event).

0x08 – The user initiated a circuit group block (CGB) or block (BLO) message, either through isup_console or through a host message (MAINT_BLOCK event).

0x09 – A circuit group reset (GRS) or reset circuit (RSC) was initiated by the host or a user through isup_console.

0x0D – A hardware-oriented circuit group unblock acknowledgement (CGUA) message was received from the SS7 network.

0x0E – A hardware-oriented circuit group block (CGB) message was received from the SS7 network.

0x0F – A hardware-oriented circuit group unblock (CGU) message was received from the SS7 network.

0x10 – A hardware-oriented circuit group blocking acknowledgement (CGBA) message was received from the SS7 network.

*NOTE: Feature Flag 16 must be enabled in the **CktInt.cfg** file to receive a $D3 report with this value.*

0x11 – A maintenance-oriented circuit group blocking acknowledgement (CGBA) message, or a blocking acknowledgement (BLA) message, was received from the SS7 network.

*NOTE: Feature Flag 16 must be enabled in the **CktInt.cfg** file to receive a $D3 report with this value.*

Resource Group (byte offset 9) – specifies the resource group number to which the port belongs; convert byte from hex to decimal for the group number (1 to 32 inclusive).

Reserved (byte offsets 10 to 12) – These bytes must be set to 00 00 00.

6.5.8 SS7 Circuit Group Status ($D9) Report (Standard)

Report Type: Circuit Group Status

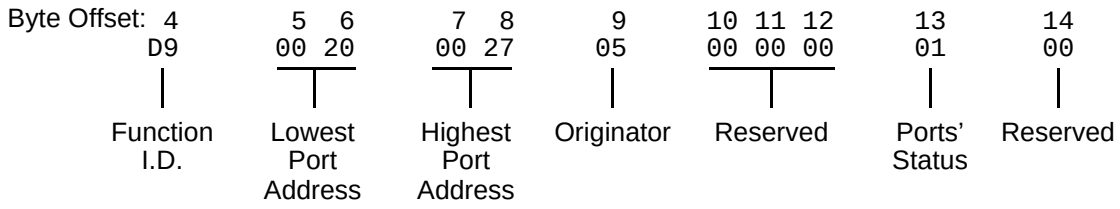
Description:

Informs the host of a change in the state of a circuit group.

Action Causing Report Generation:

Generated when cktint receives an incoming CGB/CGU/GRS from the network, system, or isup_console.

Format:



Function I.D. (byte offset 4)—Byte immediately following the Network Header; uniquely identifies the report from Circuit Interworking system.

Lowest Port Address (byte offsets 5 and 6)—Hex representation of the first port address of the circuit group for which the report is sent.

Highest Port Address (byte offsets 7 and 8)—Hex representation of the last port address on the card for which the report is sent.

Originator (byte offset 9)—Specifies whether the CGB/CGU/GRS message was originated by the system, host, or isup_console. The following originator code values are added to the CktInt $D9 reports sent to the host during SS7 events:

- 0x01 – A maintenance-oriented circuit group block (CGB) message was received from the SS7 network
- 0x02 – A maintenance-oriented circuit group unblock (CGU) message was received from the SS7 network
- 0x03 – A circuit group unblock acknowledge (CGUA) message was received from the SS7 network
- 0x04 – A circuit group reset (GRS) message was received from the SS7 network
- 0x05 – A circuit group reset acknowledge (GRA) message was received from the SS7 network
- 0x06 – A hardware-oriented circuit group block (CGB) message was sent to the SS7 network (as the result of a $D9 OOS report from the VCO)
- 0x07 – A maintenance-oriented circuit group block (CGB) message was sent to the SS7 network from the isup_console

0x08 – A circuit group reset (GRS) message was sent to the SS7 network

0x0A – A hardware-oriented circuit group block (CGB) message was received from the SS7 network.

0x0B – A hardware-oriented circuit group unblock (CGU) message was received from the SS7 network.

0x0C – A hardware-oriented circuit group unblock acknowledgement (CGUA) message was received from the SS7 network.

0x10 – A hardware-oriented circuit group blocking acknowledgement (CGBA) message was received from the SS7 network.

NOTE: Feature Flag 16 must be enabled in the ***CktInt.cfg*** *file to receive a $D9 report with this value.*

0x11 – A maintenance-oriented circuit group blocking acknowledgement (CGBA) message, or a blocking acknowledgement (BLA) message, was received from the SS7 network.

NOTE: Feature Flag 16 must be enabled in the ***CktInt.cfg*** *file to receive a $D9 report with this value.*

Reserved (byte offsets 10 to 12)—Always 00 00 00.

Ports' State (byte offset 13)—Indicates the present state of the ports for which the report is generated. Interpret this byte according to the list that follows:

01 – Available

02 – Unavailable

03 – Transient

Reserved (byte offset 14) – Always 00.

6.5.9 SS7 Network Message Reception ($EA) Report (Standard)

Report Type: Resource Control

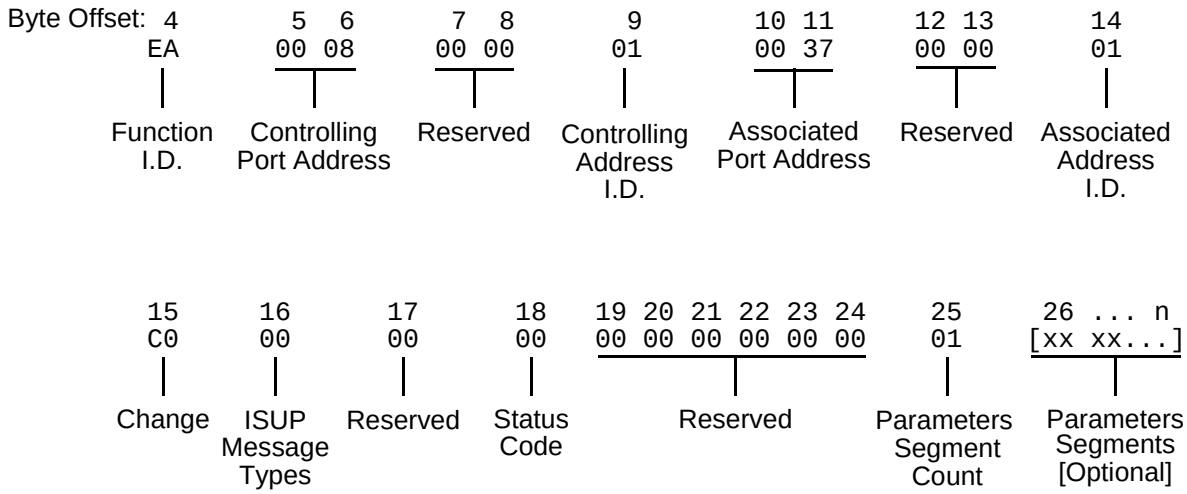
Description:

The SS7 $EA Report informs the host of a change in the state of an SS7 call. A change of state occurs when an SS7 ISUP message is received from the SS7 network. Both the affected SS7 controlling and associated ports are represented.

Ports identified in this report are SS7 ports only. Non-SS7 ports are never represented in an SS7 $EA report. (i.e. ACM received from the network with one SS7 port and one non-SS7 port—only the SS7 port affected by the ACM will be represented in the report.)

This report may be truncated if the network header segment, base report, and parameters exceed 255 bytes in length. No indication of truncation is provided to the host.

Format:



Function I.D. (byte offset 4) – Byte immediately following the Network Header; uniquely identifies the report from Circuit Interworking system.

Controlling Port Address (byte offsets 5 and 6) – Hex representation of the controlling port address for which the report is sent. These bytes represent the specific circuit assigned to the call.

Reserved (byte offsets 7 and 8) – Always 00 00.

Controlling Address Identifier (byte offset 9) – Specifies the controlling port is identified by port address. Interpret the byte according to the list below:

01 — controlling port specified by port address.

Associated Port Address (byte offsets 10 and 11) – Hex representation of the associated (outgoing) port address for which the report is sent. These bytes represent the specific circuit used.

Reserved (byte offsets 12 and 13) – Always 00 00.

Associated Address Identifier (byte offset 14) – Specifies the associated (outgoing) port is identified by port address. Interpret the byte according to the list below:

00 – No associated port

01 – Associated port specified by port address

Change (byte offset 15) – Type of change detected. The list below provides a general indication of the change that occurred for the outgoing port.

00 – No change; report issued to send SS7 message

40 – Port became inactive (RLC, REL message)

80 – Port became active (IAM received)

ISUP Message Type (byte offset 16) – Indicates the SS7 message received. The message type is the hex value for the message. Refer to Appendix C for a list of the ISUP messages and their function.

Reserved (byte offset 17) – Always 00.

Status Code (byte offset 18) – Indicates if an error was encountered. Interpret the byte according to the list below:

01 – No error

A4 – Protocol violation

Reserved (byte offset 19) – Always 00.

Reserved (byte offsets 20 to 24) – Always 00 00 00 00 00.

Parameter Segment Count Byte

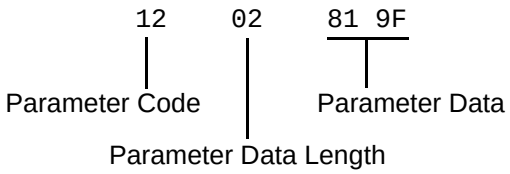
The Parameter Segment Count byte is used in all $EA reports and specifies how many parameters are included in the report.

Parameter Segment Count (byte offset 25) – Specifies how many parameters are included in the command. Convert from decimal to hex for use in the command. A value of $00 indicates that no parameters follow this byte.

Parameter Segment

The Parameter Segment bytes contain parameters to be received from the SS7 network. The number of parameters contained in this segment appears in the Parameter Segment Count byte.

Each parameter consists of a Parameter Code byte, a Parameter Data Length byte, and one or more Parameter Data bytes. The format and components of the Parameter Segment are defined as follows:



Parameter Code — Specifies the parameter code.

> *NOTE: When the AOC feature is being used, this will be specified as APP (Application Transport Parameter—0x78). See Section 6.7 for additional information on special coding for this parameter.*

Parameter Data Length — Specifies the number of parameter data bytes; length of 00 indicates no parameter data.

Parameter Data — Specifies the parameter data exactly as it is received from the SS7 network.

6.5.10 SS7 Alarm Condition ($F0) Report (Standard)

Report Type: System Status

Description:

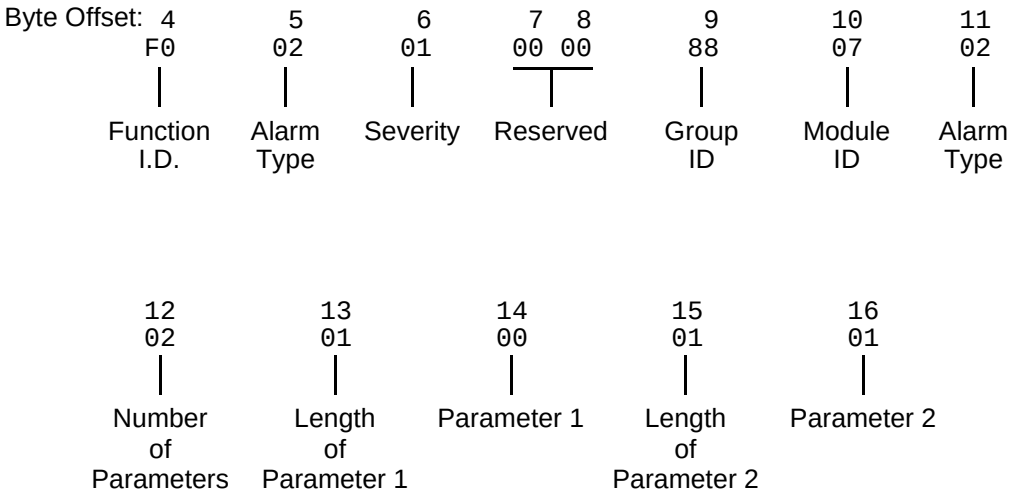
Informs the host of the presence of specified system alarms, including alarm severity and type.

The SS7 $F0 report allows the host the flexibility to take corrective action from a remote facility.

Action Causing Report Generation:

Circuit Interworking generates this report for the host when it detects link alarms—Link Unavailable (880702) or Link Available (880704)—at the MTP level. See the *NewNet AccessManager Maintenance Manual* for more information on specific alarms.

Format:



Function I.D. (byte offset 4) – Byte immediately following the Network Header; uniquely identifies the report from Circuit Interworking system.

Alarm Type (byte offset 5) – Specifies what type of alarm the system sent.

02 – Link unavailable

04 – Link available

Severity (byte offset 6) – Specifies the alarm level.

00 – Information

01 – Minor

Reserved (byte offsets 7 and 8) – Always 00 00.

Group ID (byte offset 9) – Always 88.

Module ID (byte offset 10) – Always 07.

Alarm Type (byte offset 11) – Specifies what type of alarm the system sent (same as byte offset 5).

02 – Link unavailable

04 – Link available

NOTE: Together, byte offsets 9, 10, and 11 form an alarm number described in the Alarms and Errors section of the NewNet AccessManager Maintenance Manual.

Number of Parameters. (byte offset 12) – Specifies how many parameters are included in the report.

Length of Parameter 1 (byte offset 13) – Length of Parameter 1 in bytes.

Parameter 1 (byte offset 14) – Linkset number. Linkset numbers are derived by the order in which they appear in the MTP MML file, starting with 0.

Length of Parameter 2 (byte offset 15) – Length of Parameter 2 in bytes.

Parameter 2 (byte offset 16) – Link number (same as the SLC number).

6.6 CIRCUIT INTERWORKING COMMANDS AND REPORTS (EXTENDED)

This section describes the commands and reports used by the SS7 subsystem in Extended Operational Mode. Refer to the *Programming Reference* for additional information on non-SS7 commands and reports used by Circuit Interworking.

6.6.1 SS7 Circuit Query ($30 01) Command (Extended)

Command Type: System Status

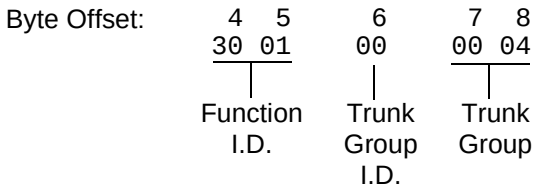
Description:

Initiates a query of the SS7 circuit states for one or all trunk groups. An SS7 Circuit Status ($B0 01) report is generated back to the host.

Usage Guidelines:

You may use this command at any time, but it is recommended that its use be restricted to periods of low traffic.

Format:



Function ID (byte offsets 4 and 5) – Byte immediately following the Network Header. The Function ID uniquely identifies the command to the SS7 subsystem.

Trunk Group ID (byte offset 6) – Specifies the trunk group to query. Specify the byte according to the list below:

00 – Query trunk group specified in byte offsets 7 and 8

01 – Query all trunk groups.

Trunk Group (byte offsets 7 and 8) – Specifies trunk group. If “all trunk groups” are specified in byte offset 6, must be set to 00 00.

Network Status Byte:

The following network status bytes are returned by the $30 01 Command:

Status Byte	Meaning
$01	Command processing successful. Indicates command sent to the system has been processed successfully. Returned only when Return All is specified in the command's Network Control byte. No corrective action required.
$D0	Invalid Trunk Group ID.
$41	Invalid host message length.

6.6.2 SS7 Circuit Sync ($30 02) Command (Extended)

Command Type: Configuration Control

Description:

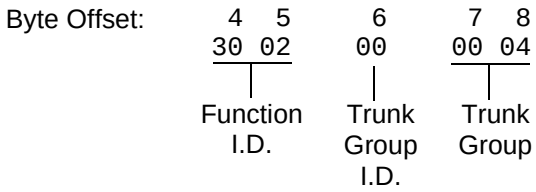
Synchronizes the CktInt circuit states of one or all trunk groups with the underlying ISUP layer.

Usage Guidelines:

This command may be used at anytime, but it is recommend that its use be restricted to periods of low traffic.

This command can be used in case the need arises to make sure the circuit states are consistent with what is maintained in the ISUP layer.

Format:



Function ID (byte offsets 4 and 5) – Byte immediately following the Network Header. The Function ID uniquely identifies the command to the SS7 subsystem.

Trunk Group ID (byte offset 6) – Specifies the trunk group to sync. Specify the byte according to the list below:

00 – Sync trunk group specified in byte offsets 7 and 8

01 – Sync all trunk groups.

Trunk Group (byte offsets 7 and 8) – Specifies trunk group. If “all trunk groups” are specified in byte offset 6, must be set to 00 00.

Network Status Byte:

The following network status bytes are returned by the $30 02 Command:

Status Byte	Meaning
$01	Command processing successful. Indicates command sent to the system has been processed successfully. Returned only when Return All is specified in the command's Network Control byte. No corrective action required.
$D0	Invalid Trunk Group ID.
$41	Invalid host message length.

6.6.3 SS7 Network Message Generation ($49) Command (Extended)

Command Type: Resource Control

Description:

The SS7 $49 Command is used for all call control functions for an SS7 call, including connecting an incoming and outgoing port, and forcing SS7 call origination. Non-SS7 network interface ports can also be controlled with this command in an interworking scenario. The $49 Command functions as follows:

Set Disconnect Control – Each SS7 $49 Command for an SS7 controlled port sets the Disconnect Control values. Therefore the command must specify the desired Disconnect Control values. This allows it to be changed after a call has been established.

Create Call – To allocate an internal call object for maintaining connection relationships and take the SS7 controlled port(s) off-hook, you must set the Switching and Attaching bits in the Connection Control byte in the first SS7 $49 for the call.

Release Call – To deallocate the internal call object and put the SS7 controlled port(s) on-hook, you must set the Switching bit in the Connection Control byte of the $49 Command releasing the call. Do not set the Attaching bit.

Associate Ports – To make the system consider the Disconnect Control values in a $49 Command for a non-SS7 controlled port, associate the port with an SS7 controlled port in the call object. You can do this at anytime by providing the non-SS7 port as the other Port Address, either Controlling or Associated.

Send ISUP Message – Set the Template Control byte or the ISUP Message Type byte to send an ISUP message through the Associated Port Address.

- If both the Controlling and Associated Port Addresses are not SS7 controlled or not specified, then network status byte $03 is returned.
- If both the Template Control byte and the ISUP Message Type byte are zero (0) then no ISUP message is sent.

Usage Guidelines:

To use the SS7 $49 Command to generate an ISUP message, the destination VCA must be set to $C0, and either byte offset 17 (ISUP Message Type) must be set for the message type, or byte offset 31 (Template Control) must be set to the template number.

Use this command for SS7 call control functions, including connecting an incoming and outgoing port, and forcing SS7 call origination. Control over the contents of SS7-generated messages is provided.

Total command length cannot exceed 306 bytes.

The controlling port is specified by port address.

A virtual port is not required to originate an outgoing call. Set the Control Address ID to 0, if you want the Controlling Port Address to be used by the host for tracking purposes.

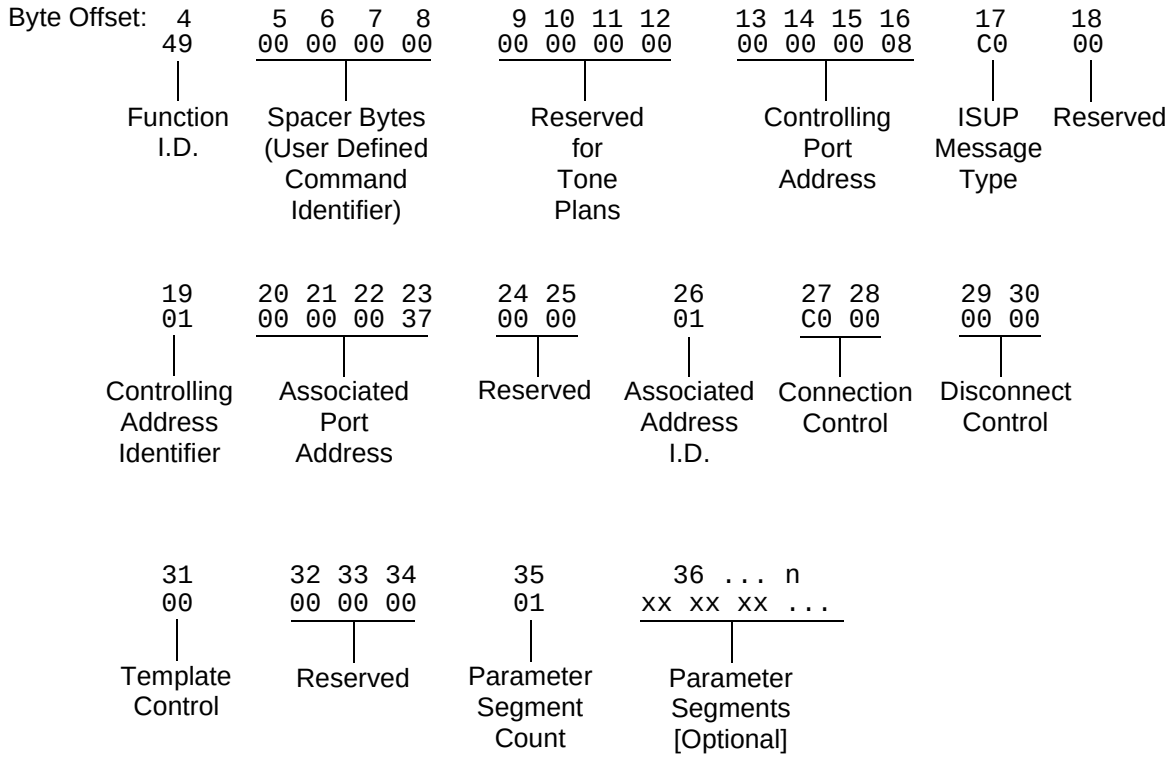
You can choose the associated (outgoing) port two ways: specify by port address or hunt through a resource group.

The returned command is truncated to report only through byte offset 26 (Associated Address I.D.).

In interworking scenarios, either the controlling or associated port can be a non-SS7 port.

If you specify parameters, their values must be coded as they would appear in an SS7 message. Parameters you specify in the command are included in the next SS7 message transmission. The host-specified parameters are added to any SS7 parameters included in the template being executed.

Format:



Function I.D. (byte offset 4) – Byte immediately following the Network Header; uniquely identifies the command to Circuit Interworking.

Spacer Bytes (User Defined Command Identifier) (byte offsets 5 to 8) – Reserved for the host. Since these are echoed back in the command response, you can use them to reference or "name" calls.

Reserved for Tone Plans (byte offsets 9 to 12) – Reserved for future tone plan specifications.

Controlling Port Address (byte offsets 13 to 16) – Hex representation of the controlling port circuit address for which the command is sent.

ISUP Message Type (byte offset 17) – The ISUP messages to generate only if the Template Control byte (byte offset 31) is set to zero.

Reserved (byte offset 18) – Always 00

Controlling Address I.D. (byte offset 19) – Specify the byte according to the following:

00 – no controlling address used in command. Controlling Port Address may be used by the host for tracking purposes. (To initiate outbound SS7 calls, set to 00.)

01 – controlling port specified by port address

Associated Port Address (byte offsets 20 to 23) – Hex representation of the associated (outgoing) port circuit address for which the command is sent or the resource group number from which to select the outgoing port. The Associated Address I.D. (byte offset 26) must be set to $01.

Reserved (byte offset 24 and 25) – These bytes must be set to 00 00.

Associated Address I.D. (byte offset 26) – Specifies that the associated (outgoing) port is identified by port address/resource group. Specify the byte according to the list below:

00 – no associated address used in command

01 – associated port specified by port address, or resource group number

Connection Control (byte offsets 27 and 28) – Specifies the switching, attaching, and hunting options when this command is used to connect a call. Byte offset 28 is reserved for future enhancements and must be set to $00. Construct byte offset 27 according to the descriptions below, then convert to hex for use in the command.

SAPVV000

S – Specifies if switching action is required

S = 0 – No switching action required; A bit ignored and should set to 0.

S = 1 – Switching action required.

A – Specifies whether to link or remove a resource

A = 0 – If S = 1, remove resource from call; if S = 0, no meaning.

A = 1 – If S = 1, link resource to call; if S = 0, no meaning.

P – Specifies whether to use a specific outgoing circuit or to select any outgoing circuit from a resource group; port address or group number is specified in Associated Port Address bytes

P = 0 – For S = 0 or 1, use port specified in Associated Port Address bytes.

P = 1 – For S = 1, select port from resource group specified in Associated Port Address bytes; the port address of the selected channel is specified in the command returned to the host.

VV – Not used for SS7. Set to 00.

Disconnect Control Bytes

The Disconnect Control bytes are always included in the command. You can only define byte offset 29 at this time. Byte offset 29 specifies the disposition of ports when the call is torn down. Byte offset 30 is reserved for future enhancements and should be set to $00.

Bit settings in this byte are overridden if another $49 Command is processed for either port in this $49 Command.

Disconnect Control (byte offset 29 and 30) – Determines what actions to take on a port when the opposite end goes on hook. Construct byte in binary according to the descriptions below, then convert to hex for use in the command.

0000I0U0

I – Specifies whether to return the incoming port to CP_SETUP state when the outgoing port goes on hook

I = 0 – force incoming to idle

I = 1 – set incoming to setup state upon outgoing disconnect; T must be 0

U – Specifies whether to return the outgoing port to CP_SETUP state when the in coming port goes on hook

U = 0 – force outgoing to idle

U = 1 – set outgoing to setup state upon incoming disconnect; C must be 0

Template Control Byte

The Template Control byte is always included in the command, and specifies the template to be used when transmitting an SS7 message.

Template Control (byte offset 31) – Specifies whether a template is to be used in this command and the template number, if any (see *Section 6.2*). Construct byte in binary according to the descriptions that follow, then convert to hex for use in the command.

X0RRRRRR

X – Specifies if an template is used in this command

X = 0 – no template specified. Use ISUP Message Type byte (offset byte 7).

X = 1 – execute template specified in RRRRRR; I must be 0

RRRRRR – Specifies the template number.

Reserved (byte offsets 32 to 34) – Always 00 00 00.

Parameter Segment Count Byte

The Parameter Segment Count byte is used in all $49 Commands and specifies how many parameters are included in the command.

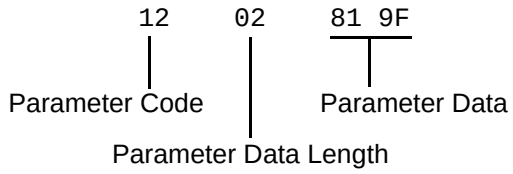
Parameter Segment Count (byte offset 35) – Specifies how many parameters are included in the command. Convert from decimal to hex for use in the command. A value of 00 indicates that no parameters follow this byte.

Parameter Segment

The Parameter Segment bytes contain parameters to be transmitted to the SS7 network. The number of parameters contained in this segment appears in the Parameter Segment Count byte.

> *NOTE: Each time a $49 command is sent, the parameters defined in that particular command override any permanent parameters set in the template it names in byte offset 31.*

Each parameter consists of a Parameter Code byte, a Parameter Data Length byte, and one or more Parameter Data bytes. The format and components of the Parameter Segment are defined as follows:



Parameter Code – Specifies the parameter code.

> *NOTE: When using the AOC feature, this must be specified as APP (Application Transport Parameter—0x78). See Section 6.7 for additional information on special coding for this parameter.*

Parameter Data Length – Specifies the number of parameter data bytes.

Parameter Data – Specifies the parameter data exactly as it should be sent to the SS7 network.

Coding for Pass Along Message (PAM)

End-to-End signalling uses the pass along method of signalling in which information is exchanged between two end points via the Pass Along Message (PAM). The contents of this message are relevant only to the end points; all other intermediate exchanges just pass it along transparently.

PAM is characterized by a special message type code as specified in ITU Recommendation Q.763. It has a value of 0x28. One PAM message may have one embedded ISUP message to be passed along.

Byte offset 17 of the $49 command must be set to 0x28 when a PAM message is sent. Then, the coding from byte offset 35 onward is shown in Figure 6.7.

This example shows the parameter segment count and parameter segment for a PAM with a Call Progress Message (CPG) as the embedded ISUP message.

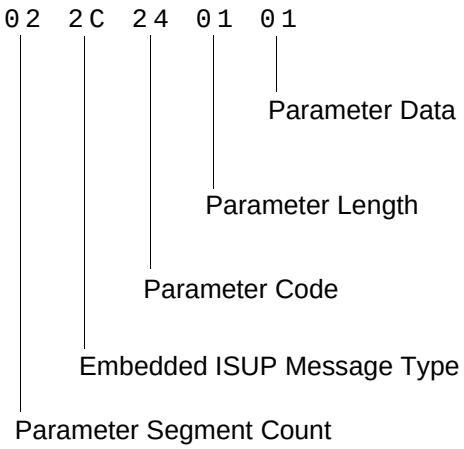


Figure 6.8: Example of CPG Message Embedded in a PAM

Parameter Segment Count – Specifies the number of parameters in the embedded ISUP message + 1 [the extra count is for the message type of the embedded ISUP message].

Embedded ISUP Message Type – Specifies the message type of the embedded ISUP message; no associated length and data information will follow.

Parameter Code – Specifies the parameter code of the parameter in the embedded ISUP message.

Parameter Length – Specifies the number of parameter data bytes.

Parameter Data – Specifies the parameter data exactly as it should be sent to the SS7 network.

Network Status Byte:

The following network status bytes are returned by the $49 Command:

Status Byte	Meaning
$01	Command processing successful. Indicates command sent to the system has been processed successfully. Returned only when Return All is specified in the command's Network Control byte. No corrective action required.
$03	Syntax error in command. Indicates one or more of the values specified in the command are invalid or that the command specifies no action or improper switching actions within the SAP bit settings. Specifying values in spacer bytes or reserved bytes may also cause syntax errors. Check all byte values in command.
$08	Command received was received by Standby side but can only be processed on the Active side.
$0D	Invalid resource group number. The resource group specified in the command is 0 or greater than 224. Check resource group number.
$0E	Invalid controlling port address (not in valid range). The port address specified in the command is not within the range of system port addresses or that port address is not assigned. Check port address.
$0F	Invalid controlling host
$15	Invalid associated port address (not in valid range). Indicates the port address specified in the command is not within the range of system port addresses. Check port address or select a port from appropriate resource group.
$2B	The Template number specified in the command is out of the range for a rule (1 to 32). Check value and resend command with correct value.
$41	Invalid host message length.
$C7	Invalid range specified in group maintenance messages.
$C8	Incoming port is in the wrong state.
$C9	No resource available.
$CA	Outgoing port is in the wrong state.
$CB	Parameter error in the host command.
$CC	Parameter error in the template file.
$CD	Invalid ISUP message type.
$CE	Missing Range and Status parameter.
$CF	Protocol violation. This network status byte usually indicates that the ISUP message type is invalid for the current call state.

Status Byte	Meaning
$D1	The Far End SP is unavailable.
SD4	Circuit in transient state. When a circuit is in maintenance transient state, no messages are allowed on that circuit.
$D5 (AOC only)	Invalid SLR (Segmentation Local Reference). Indicates the host application sent charging information segmented into two or more $49 commands and there was a mismatch in the outgoing SLR between the segments. Thus, the host command(s) were rejected by cktint.
$D6 (AOC only)	Invalid host cktint SI (Segment Indicator). Indicates the host application sent charging information segmented into two or more $49 commands and there was a mismatch in the outgoing SI between the segments. Thus, the host command(s) were rejected by cktint.

6.6.4 SS7 Host Call Load Control ($C0 04) Command (Extended)

Command Type: Configuration Control

Description:

Used in conjunction with the Host Control of Call Load feature in the Generic software (refer to the System Host Configuration screen in the *System Administrator's Guide*). When the feature is enabled, this command allows the host to indicate its ability to process all calls, existing calls only, or no calls.

Usage Guidelines:

The system verifies that the Host Control of Call Load feature has been enabled before processing the command. If the feature is not enabled in the Generic software, the command is returned with a network status byte of $1B (feature not enabled) in the network header.

If the load control code is set to 00, normal call processing occurs with no restrictions.

If the load control code is set to 01, cktint does not send SS7 $EA reports to the host or process SS7 $49 commands containing call control messages from the host until the host sends an SS7 $C0 04 command indicating that it is ready to process calls. All maintenance reports are still sent to the host; all maintenance message commands are also sent and processed.

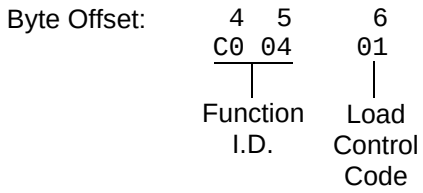
In addition, if the load control code is set to 02, cktint does not send new call reports to the host or process commands containing IAM messages from the host until the host indicates that it is ready to process new calls. However, commands will still be accepted from and reports sent to the host for calls in progress.

You can use the SS7 $C0 04 command various ways. For example, sending an SS7 $C0 04 command with the load control code set to 01 would allow the host to perform some type of initialization before accepting calls (the default setting on startup when the Host Control of Call Load feature is enabled). With the load control code set to 02, current calls will be completed, but no new calls accepted—a "graceful shutdown."

This command only affects operations with respect to the host from which it is received. If there are multiple hosts, the command must be sent by each host.

> *NOTE:* ***On an SS7 system, the $C0 04 command destination VCA must be set to $C0****; otherwise, this could result in tcp link state mismatches between the VCO/4K, cktint, and the host.*

Format:



Function ID (byte offset 4 and 5) – Byte immediately following the Network Header. The Function ID uniquely identifies the command to the SS7 subsystem.

Load Control Code (byte offset 6) – Determines whether the system should process incoming calls by reporting them to the host or reject call control commands sent by the host. Specify this byte according to the following list:

00 = Normal call processing

01 = Call processing suspended

02 = No new calls accepted or sent from the host; process existing calls only

Network Status Byte:

The following network status bytes are returned by the SS7 $C0 04 Command:

Status Byte	Meaning
$01	Command processing accepted. Indicates command sent to the system has passed initial processing. Returned only when Return All is specified in the command's Network Control byte. No corrective action required.
$03	Syntax error in command.
$1B	Host Control of Call Load feature is not enabled and command was not processed.
$D2	VCO/4K is not connected and command was not processed.

Host Link States:

The following table summarizes possible Host Link States associated with the SS7 $C0 04 Command:

Host Link State	Affect on Messaging
ONLINE_READY	Normal messaging. All host commands and reports allowed. Set by SS7 $C0 04 host command with the load control code set to 00.
ONLINE_NOT_READY	Call processing effectively suspended; all call control commands rejected. • No new SS7 $EA call reports sent. • No call control commands allowed; only configuration commands (i.e., $C0 04, $C0 01, $30 01, $30 02) and maintenance messages. Set by SS7 $C0 04 host command with the load control code set to 01.
ONLINE_RESTRICTED	Processing of calls in progress; host commands accepted. • No SS7 $EA report for new call reports for incoming IAM. • All host commands allowed, except SS7 $49 command with IAM. Set by SS7 $C0 04 host command with the load control code set to 02.

NOTE: If calls are present when the host goes ONLINE_NOT_READY, there can be a mismatch of circuit states among the VCO/4K, cktint, and the host.

When the Host Control of Call Load feature is enabled and there is a connection to the host, the initial tcp link state in cktint is ONLINE_NOT_READY. *It stays in this state until receiving an SS7 $C0 04 command from the host requesting a host link state change.*

6.6.5 SS7 Host Assume/Relinquish Port Control ($C0 05) Command (Extended)

Command Type: Configuration Control

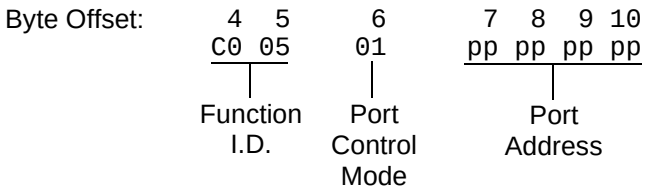
Description:

Allows a host process to relinquish control of a call assigned to itself, or assume control of a call that has been assigned to a different host.

Usage Guidelines:

This command performs the assume/relinquish port control function. A function code allows a host link to override a controlling host assignment with one of two operating modes: one mode allows a host to assume control of a port; the other mode allows a host to relinquish control of a port. In either case, the assignment override remains in effect as long as the affected port is involved in a call, or until a subsequent override command is received. The relinquish mode clears the controlling host assignment for the specified port so that another host can take control of that port.

Format:



Function ID (byte offset 4 and 5) – Byte immediately following the Network Header. The Function ID uniquely identifies the command to the SS7 subsystem.

Port Control Mode (byte offset 6) – Determines the type of action to be taken. Specify this byte according to the following list:
00 = Relinquish control
01 = Assume control

Port Address (byte offsets 7 to 10) – Port address for which control is being seized or relinquished.

Network Status Byte:

The following network status bytes are returned by the $C0 05 Command:

Status Byte	Meaning
$01	Command processing accepted. Indicates command sent to the system has passed initial processing. Returned only when Return All is specified in the command's Network Control byte. No corrective action required.
$03	Syntax error in command. Indicates one or more of the values specified in the command are invalid or that the command specifies no action or improper switching actions within the SAP bit settings. Specifying values in spacer bytes or reserved bytes may also cause syntax errors. Check all byte values in command.
$0E	Invalid controlling port address (not in valid range). Indicates the port address specified in the command is not within the range of system port addresses or that port address is not assigned. Check port address.
$C9	No resource available.

6.6.6 SS7 Circuit Status ($B0 01) Report (Extended)

Report Type: System Status

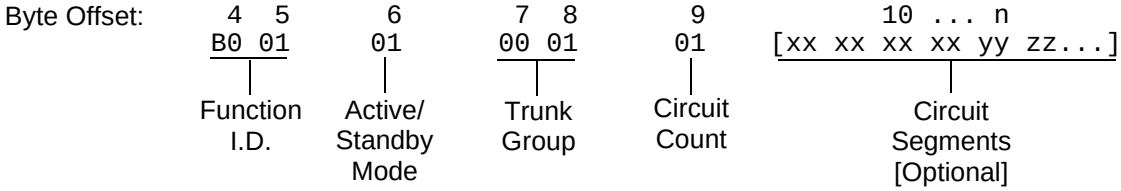
Description:

Reports the SS7 circuit states for a trunk group.

Action Causing Report Generation:

Generated in direct response to an SS7 Circuit Query ($30 01) Command. If all trunk groups were requested, then multiple reports are generated, one for each trunk group.

Format:



Main Command Segment

Function ID (byte offsets 4 and 5) – Bytes immediately following the Network Header; uniquely identifies this report from Circuit Interworking.

Active/Standby Mode (byte offset 6) – Indicates active/standby mode. Non-redundant systems will always return active mode.

0 = Standby mode

1 = Active mode

Trunk Group (byte offsets 7 and 8) – Specifies the trunk group.

Circuit Count (byte offset 9) – Indicates the number of circuit segments that follow. This will be a value from 0 to 32.

Circuit Segments

Circuit Segments (byte offset 10 – n) – Each segment consists of six bytes. They are the Circuit Port Address (4 bytes), the Circuit State/Call State (1 byte), and the Circuit Sub State (1 byte). The number of circuit segments is specified by the Circuit Count.

Circuit Port Address – The VCO/4K port address associated with this circuit.

Circuit State/Call State — The (maintenance) state of this circuit. The possible hex values are as follows:

FF Unknown
00 Unequipped
01 Transient
10 Unequipped/Idle

12 Active/Idle
13 Remotely Blocked/Idle
14 Locally Blocked/Idle
15 Locally and Remotely Blocked/Idle
16 Locally Block_Transient/Idle
17 Locally Unblock_Transient/Idle
18 Locally Block_Transient and Remotely Blocked /Idle
19 Locally Unblock_Transient and Remotely Blocked /Idle

22 Active/Incoming Busy
23 Remotely Blocked/Incoming Busy
24 Locally Blocked/Incoming Busy
25 Locally and Remotely Blocked/Incoming Busy
26 Locally Block_Transient/Incoming Busy
27 Locally Unblock_Transient/Incoming Busy
28 Locally Block_Transient and Remotely Blocked/Incoming Busy
29 Locally Unblock_Transient and Remotely Blocked/Incoming Busy

32 Active/Outgoing Busy
33 Remotely Blocked/Outgoing Busy
34 Locally Blocked/Outgoing Busy
35 Locally and Remotely Blocked/Outgoing Busy
36 Locally Block_Transient/Outgoing Busy
37 Locally Unblock_Transient/Outgoing Busy
38 Locally Block_Transient and Remotely Blocked/Outgoing Busy
39 Locally Unblock_Transient and Remotely Blocked/Outgoing Busy

82 Hardware Remotely Blocked, Maintenance Active/Idle
83 Hardware Remotely Blocked, Maintenance Remotely Blocked/Idle
84 Hardware Remotely Blocked, Maintenance Locally Blocked/Idle

85 Hardware Remotely Blocked, Maintenance Locally and Remotely Blocked/Idle

86 Hardware Remotely Blocked, Maintenance Locally Block_Transient/Idle

87 Hardware Remotely Blocked, Maintenance Locally Unblock_Transient/Idle

88 Hardware Remotely Blocked, Maintenance Locally Block_Transient and Remotely Blocked/Idle

89 Hardware Remotely Blocked, Maintenance Locally Unblock_Transient and Remotely Blocked/Idle

92 Hardware Locally Blocked, Maintenance Active/Idle

93 Hardware Locally Blocked, Maintenance Remotely Blocked/Idle

94 Hardware Locally Blocked, Maintenance Locally Blocked/Idle

95 Hardware Locally Blocked, Maintenance Locally and Remotely Blocked/Idle

96 Hardware Locally Blocked, Maintenance Locally Block_Transient/Idle

97 Hardware Locally Blocked, Maintenance Locally Unblock_Transient/Idle

98 Hardware Locally Blocked, Maintenance Locally Block_Transient and Remotely Blocked/Idle

99 Hardware Locally Blocked, Maintenance Locally Unblock_Transient and Remotely Blocked/Idle

A2 Hardware Locally and Remotely Blocked, Maintenance Active/Idle

A3 Hardware Locally and Remotely Blocked, Maintenance Remotely Blocked/Idle

A4 Hardware Locally and Remotely Blocked, Maintenance Locally Blocked/Idle

A5 Hardware Locally and Remotely Blocked, Maintenance Locally and Remotely Blocked/Idle

A6 Hardware Locally and Remotely Blocked, Maintenance Locally Block_Transient/Idle

A7 Hardware Locally and Remotely Blocked, Maintenance Locally Unblock_Transient/Idle

A8 Hardware Locally and Remotely Blocked, Maintenance Locally Block_Transient and Remotely Blocked/Idle

A9 Hardware Locally and Remotely Blocked, Maintenance Locally Unblock_Transient and Remotely Blocked/Idle

B2 Hardware Locally Block_Transient, Maintenance Active/Idle

B3 Hardware Locally Block_Transient, Maintenance Remotely Blocked/Idle

B4 Hardware Locally Block_Transient, Maintenance Locally Blocked/Idle

B5 Hardware Locally Block_Transient, Maintenance Locally and Remotely Blocked/Idle

B6 Hardware Locally Block_Transient, Maintenance Locally Block_Transient/Idle

B7 Hardware Locally Block_Transient, Maintenance Locally Unblock_Transient/Idle

B8 Hardware Locally Block_Transient, Maintenance Locally Block_Transient and Remotely Blocked/Idle

B9 Hardware Locally Block_Transient, Maintenance Locally Unblock_Transient and Remotely Blocked/Idle

C2 Hardware Locally Unblock_Transient, Maintenance Active/Idle

C3 Hardware Locally Unblock_Transient, Maintenance Remotely Blocked/Idle

C4 Hardware Locally Unblock_Transient, Maintenance Locally Blocked/Idle

C5 Hardware Locally Unblock_Transient, Maintenance Locally and Remotely Blocked/Idle

C6 Hardware Locally Unblock_Transient, Maintenance Locally Block_Transient/Idle

C7 Hardware Locally Unblock_Transient, Maintenance Locally Unblock_Transient/Idle

C8 Hardware Locally Unblock_Transient, Maintenance Locally Block_Transient and Remotely Blocked/Idle

C9 Hardware Locally Unblock_Transient, Maintenance Locally Unblock_Transient and Remotely Blocked/Idle

D2 Hardware Locally Block_Transient and Remotely Blocked, Maintenance Active/Idle

D3 Hardware Locally Block_Transient and Remotely Blocked, Maintenance Remotely Blocked/Idle

D4 Hardware Locally Block_Transient and Remotely Blocked, Maintenance Locally Blocked/Idle

D5 Hardware Locally Block_Transient and Remotely Blocked, Maintenance Locally and Remotely Blocked/Idle

D6 Hardware Locally Block_Transient and Remotely Blocked, Maintenance Locally Block_Transient/Idle

D7 Hardware Locally Block_Transient and Remotely Blocked, Maintenance Locally Unblock_Transient/Idle

D8 Hardware Locally Block_Transient and Remotely Blocked, Maintenance Locally Block_Transient and Remotely Blocked/Idle

D9 Hardware Locally Block_Transient and Remotely Blocked, Maintenance Locally Unblock_Transient and Remotely Blocked/Idle

E2 Hardware Locally Unblock_Transient and Remotely Blocked, Maintenance Active/Idle

E3 Hardware Locally Unblock_Transient and Remotely Blocked, Maintenance Remotely Blocked/Idle

E4 Hardware Locally Unblock_Transient and Remotely Blocked, Maintenance Locally Blocked/Idle

E5 Hardware Locally Unblock_Transient and Remotely Blocked, Maintenance Locally and Remotely Blocked/Idle

E6 Hardware Locally Unblock_Transient and Remotely Blocked, Maintenance Locally Block_Transient/Idle

E7 Hardware Locally Unblock_Transient and Remotely Blocked, Maintenance Locally Unblock_Transient/Idle

E8 Hardware Locally Unblock_Transient, and Remotely Blocked Maintenance Locally Block_Transient and Remotely Blocked/Idle

E9 Hardware Locally Unblock_Transient and Remotely Blocked, Maintenance Locally Unblock_translock_Transient and Remotely Blocked/Idle

Circuit Sub State — The call state of this circuit. The possible hex values are as follows:

00 Idle

01 Delivered

02 Present

03 Answered

04 Released

05 Received

06 Initiated

07 Continuity

08 Reserved

09 Continuity Reserved

0A Continuity Present

0B Continuity Initiated

0C Continuity Initiated Detected

0D Continuity Initiated End

0E Continuity End

0F Continuity Previous Call Present

10 Continuity Previously Initiated

11 Locally Suspended

12 Remotely Suspended

13 Local and Remote Suspended

6.6.7 SS7 System Port Status ($D3) Report (Extended)

Report Type: System Status

Description:

Informs the host of an attempted change in the status of a system resource port. Change can be the result of:

- Activating/deactivating a port or card span (refer to the *System Administrator's Guide*)
- Transmission or reception of an SS7 ISUP Maintenance message

Action Causing Report Generation:

This report is generated when a change occurs in the operating status of an individual SS7 circuit. Status changes can be caused by an action at the system administrative console (isup_console), a host command, or a received SS7 ISUP maintenance message.

A report indicating a port's availability may be returned from a circuit in Active mode. When a circuit is Out-of-Service, no other $D3 reports are generated for that port.

Format:

Byte Offset:	4	5	6 7 8 9	10	11 12	13 14 15
	D3	13	00 00 00 18	02	00 03	00 00 00
	Function I.D.	Old Status/ New Status	Port Address	Originator	Resource Group	Reserved

Function I.D. (byte offset 4) – Byte immediately following the Network Header; uniquely identifies the report from the system.

Old Status/New Status (byte offset 5) – Specifies the status of the port before the change occurred and the present status of the port. Convert byte from hex to binary and interpret the bits as described on the following page.

MMMMNNNN

MMMM – specifies the status of the port before the change occurred

MMMM = 0001 – resource was unavailable (Out of Service)

MMMM = 0011 – resource was on line and available

NNNN – specifies the current status of the port

NNNN = 0001 – resource is currently unavailable (Out of Service)

NNNN = 0011 – resource currently on line and available

Port Address (byte offsets 6 to 9) – Hex representation of port address for which the report is generated.

Originator (byte offset 10) – Specifies whether the change in status was originated by the system or the host and the reason for the change. The following originator code values are added to the CktInt $D3 reports sent to the host during SS7 events:

0x01 – A circuit group block (CGB) or a block (BLO) message was received from the SS7 network (SS7_BLOCKING event).

0x02 – A circuit group unblock (CGU) or a unblock (UBL) message was received from the SS7 network (SS7_UNBLOCKING event).

0x03 – A circuit group unblock acknowledge (CGUA) or an unblock acknowledge (UBA) message was received from the SS7 network (SS7_UNBLOCKING_ACK event).

0x04 – A reset circuit (RSC), circuit group reset (GRS), release complete (RLC), or circuit group reset acknowledge (GRA) message was received from the SS7 network (SS7_RESET event).

0x05 – A change of port status from OFFLINE to ONLINE was received from the VCO/4K in the form of $D9 or $D3 reports (SDS_UNBLOCK event).

0x06 – A change of port status to OFFLINE was received from the VCO/4K in the form of $D9 or $D3 reports (SDS_BLOCK event).

0x07 – The user initiated a circuit group unblock (CGU) or unblock (UBL) message either through isup_console or through a host message (MAINT_UNBLOCK event).

0x08 – The user initiated a circuit group block (CGB) or block (BLO) message, either through isup_console or through a host message (MAINT_BLOCK event).

0x09 – A circuit group reset (GRS) or reset circuit (RSC) was initiated by the host or a user through isup_console.

0x0D – A hardware-oriented circuit group unblock acknowledgement (CGUA) message was received from the SS7 network.

0x0E – A hardware-oriented circuit group block (CGB) message was received from the SS7 network.

0x0F – A hardware-oriented circuit group unblock (CGU) message was received from the SS7 network.

0x10 – A hardware-oriented circuit group blocking acknowledgement (CGBA) message was received from the SS7 network.

*NOTE: Feature Flag 16 must be enabled in the **CktInt.cfg** file to receive a $D3 report with this value.*

0x11 – A maintenance-oriented circuit group blocking acknowledgement (CGBA) message, or a blocking acknowledgement (BLA) message, was received from the SS7 network.

*NOTE: Feature Flag 16 must be enabled in the **CktInt.cfg** file to receive a $D3 report with this value.*

Resource Group (byte offsets 11 and 12) – specifies the resource group number to which the port belongs; convert byte from hex to decimal for the group number (1 to 32 inclusive).

Reserved (byte offsets 13 to 15) – These bytes must be set to 00 00 00.

6.6.8 SS7 Circuit Group State ($D9) Report (Extended)

Report Type: Circuit Group Status

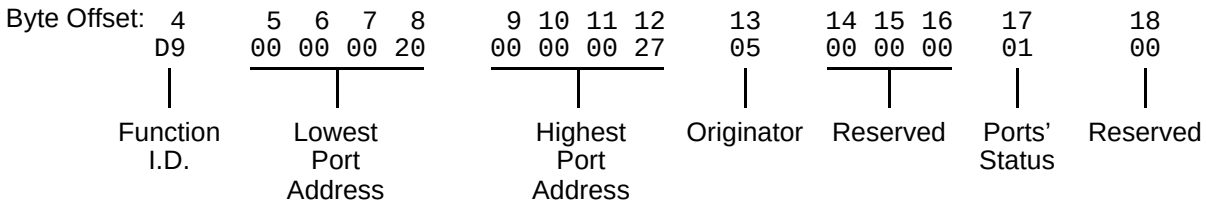
Description:

Informs the host of a change in the state of a circuit group.

Action Causing Report Generation:

Generated when cktint receives an incoming CGB/CGU/GRS from the network, system, or isup_console.

Format:



Function I.D. (byte offset 4)—Byte immediately following the Network Header; uniquely identifies the report from Circuit Interworking system.

Lowest Port Address (byte offsets 5 to 8)—Hex representation of the first port address of the circuit group for which the report is sent.

Highest Port Address (byte offsets 9 to 12)—Hex representation of the last port address on the card for which the report is sent.

Originator (byte offset 13)—Specifies whether the CGB/CGU/GRS message was originated by the system, host, or isup_console. The following originator code values are added to the CktInt $D9 reports sent to the host during SS7 events:

0x01 – A maintenance-oriented circuit group block (CGB) message was received from the SS7 network

0x02 – A maintenance-oriented circuit group unblock (CGU) message was received from the SS7 network

0x03 – A circuit group unblock acknowledge (CGUA) message was received from the SS7 network

0x04 – A circuit group reset (GRS) message was received from the SS7 network

0x05 – A circuit group reset acknowledge (GRA) message was received from the SS7 network

0x06 – A hardware-oriented circuit group block (CGB) message was sent to the SS7 network (as the result of a $D9 OOS report from the VCO)

0x07 – A maintenance-oriented circuit group block (CGB) message was sent to the SS7 network from the isup_console

0x08 – A circuit group reset (GRS) message was sent to the SS7 network

0x0A – A hardware-oriented circuit group block (CGB) message was received from the SS7 network.

0x0B – A hardware-oriented circuit group unblock (CGU) message was received from the SS7 network.

0x0C – A hardware-oriented circuit group unblock acknowledgement (CGUA) message was received from the SS7 network.

0x10 – A hardware-oriented circuit group blocking acknowledgement (CGBA) message was received from the SS7 network.

NOTE: Feature Flag 16 must be enabled in the ***CktInt.cfg*** *file to receive a $D9 report with this value.*

0x11 – A maintenance-oriented circuit group blocking acknowledgement (CGBA) message, or a blocking acknowledgement (BLA) message, was received from the SS7 network.Reserved (byte offsets 14 to 16)—Always 00 00 00.

NOTE: Feature Flag 16 must be enabled in the ***CktInt.cfg*** *file to receive a $D9 report with this value.*

Ports' State (byte offset 17)—Indicates the present state of the ports for which the report is generated. Interpret this byte according to the list that follows:

01 – Available

02 – Unavailable

03 – Transient

Reserved (byte offset 18) – Always 00.

6.6.9 SS7 Network Message Reception ($EA) Report (Extended)

Report Type: Resource Control

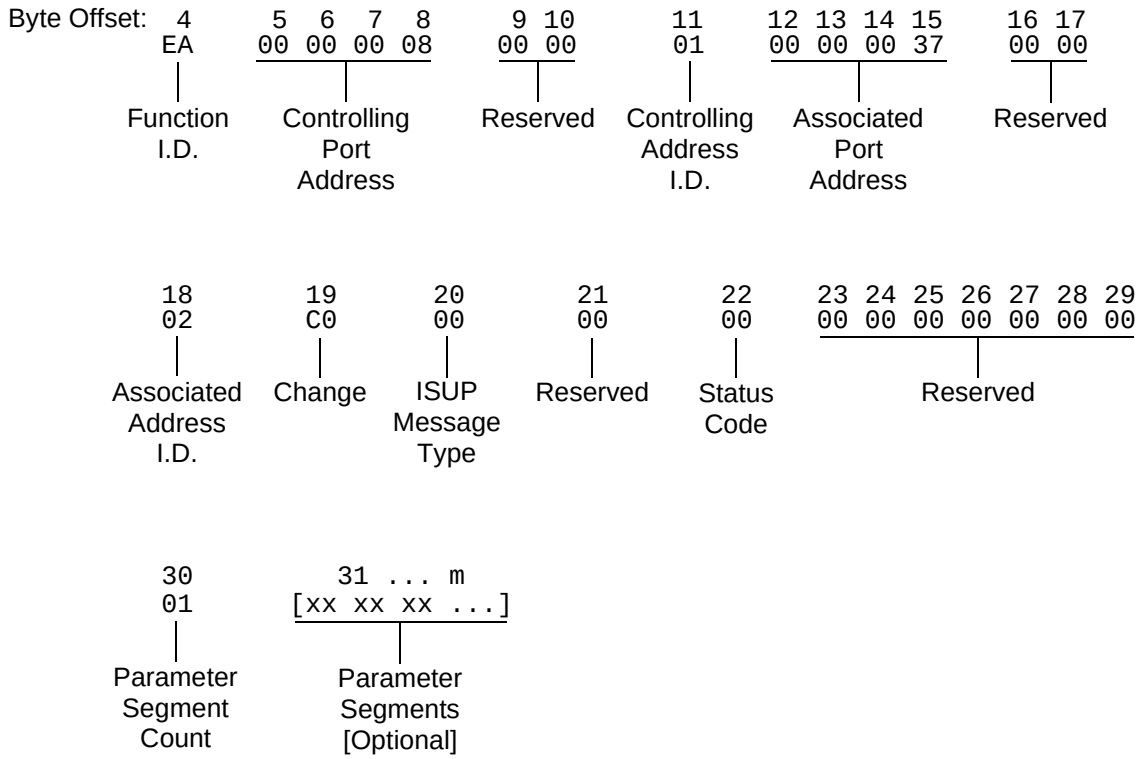
Description:

The SS7 $EA Report informs the host of a change in the state of an SS7 call. A change of state occurs when an SS7 ISUP message is received from the SS7 network. Both the affected SS7 controlling and associated ports are represented.

Ports identified in this report are SS7 ports only. Non-SS7 ports are never represented in an SS7 $EA report. (i.e. ACM received from the network with one SS7 port and one non-SS7 port—only the SS7 port affected by the ACM will be represented in the report.)

This report may be truncated if the network header segment, base report, and parameters exceed 255 bytes in length. No indication of truncation is provided to the host.

Format:



Function I.D. (byte offset 4) – Byte immediately following the Network Header; uniquely identifies the report from the system.

Controlling Port Address (byte offsets 5 to 8) – Hex representation of the controlling port address for which the report is sent. These bytes represent the specific circuit assigned to the call.

Reserved (byte offsets 9 and 10) – Always 00 00.

Controlling Address Identifier (byte offset 11) – Specifies the controlling port is identified by port address. Interpret the byte according to the list below:

01—Controlling port specified by port address.

Associated Port Address (byte offsets 12 to 15) – Hex representation of the associated (outgoing) port address for which the report is sent. These bytes represent the specific circuit used.

Reserved (byte offsets 16 and 17) – Always 00 00.

Associated Address I.D. (byte offset 18) – Specifies the associated (outgoing) port is identified by port address. Interpret the byte according to the list below:

00 – No associated port.

01 – Associated port specified by port address; Call ID = 00 00 for non-ISDN port address.

Change (byte offset 19) – Type of change detected. The list below provides a general indication of the change that occurred for the outgoing port:

00 – No change; report issued to send SS7 message

40 – Port became inactive (RLC, REL message)

80 – Port became active (IAM received)

ISUP Message Type (byte offset 20) – Indicates the SS7 message received. The message type is the hex value for the message. Refer to *Appendix E* for a list of the ISUP messages and their function.

Reserved (byte offset 21) – Always 00.

Status Code (byte offset 22) – Indicates if an error was encountered. Interpret the byte according to the list below:

01 – No error

A4 – protocol violation

Reserved (byte offsets 23 to 29) – Always 00 00 00 00 00 00 00.

Parameter Segment Count Byte

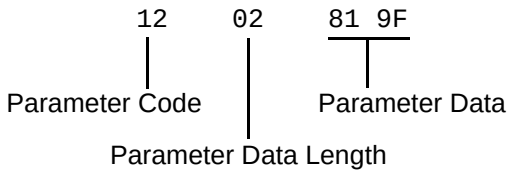
The Parameter Segment Count byte is used in all $EA reports and specifies how many parameters are included in the report.

Parameter Segment Count (byte offset 30) – Specifies how many parameters are included in the command. Convert from decimal to hex for use in the command. A value of $00 indicates that no parameters follow this byte.

Parameter Segment

The Parameter Segment bytes contain parameters to be received from the SS7 network. The number of parameters contained in this segment appears in the Parameter Segment Count byte.

Each parameter consists of a Parameter Code byte, a Parameter Data Length byte, and one or more Parameter Data bytes. The format and components of the Parameter Segment are defined as follows:



Parameter Code – Specifies the parameter code.

> *NOTE: When the AOC feature is being used, this will be specified as APP (Application Transport Parameter—0x78). See Section 6.7 for additional information on special coding for this parameter.*

Parameter Data Length – Specifies the number of parameter data bytes; length of 00 indicates no parameter data.

Parameter Data – Specifies the parameter data exactly as it is received from the SS7 network.

6.6.10 SS7 Alarm Condition ($F0) Report (Extended)

Report Type: System Status

Description:

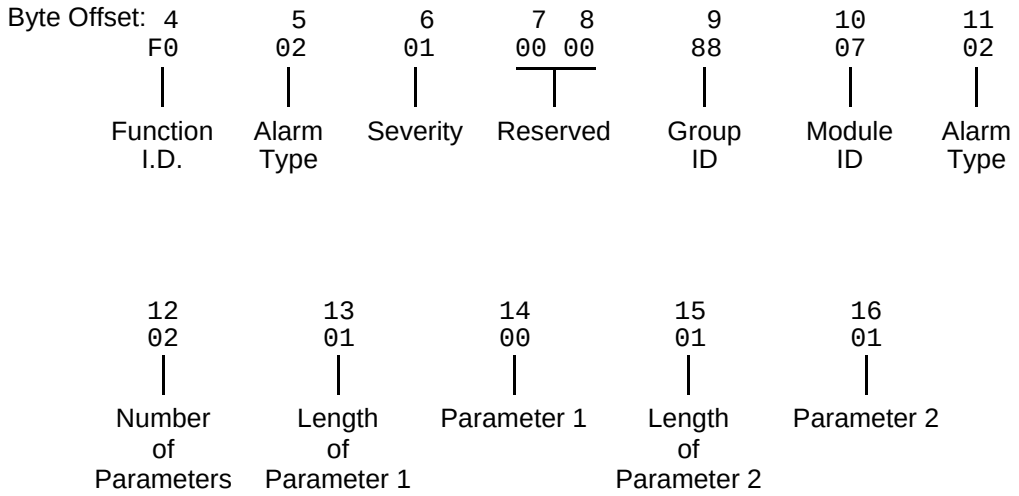
Informs the host of the presence of specified system alarms, including alarm severity and type.

The SS7 $F0 report allows the host the flexibility to take corrective action from a remote facility.

Action Causing Report Generation:

Circuit Interworking generates this report for the host when it detects link alarms—Link Unavailable (880702) or Link Available (880704)—at the MTP level. See the *NewNet AccessManager Maintenance Manual* for more information on specific alarms.

Format:



Function I.D. (byte offset 4) – Byte immediately following the Network Header; uniquely identifies the report from Circuit Interworking system.

Alarm Type (byte offset 5) – Specifies what type of alarm the system sent.

02 – Link unavailable

04 – Link available

Severity (byte offset 6) – Specifies the alarm level.

00 – Information

01 – Minor

Reserved (byte offsets 7 and 8) – Always 00 00.

Group ID (byte offset 9) – Always 88.

Module ID (byte offset 10) – Always 07.

Alarm Type (byte offset 11) – Specifies what type of alarm the system sent (same as byte offset 5).

02 – Link unavailable

04 – Link available

NOTE: Together, byte offsets 9, 10, and 11 form an alarm number described in the Alarms and Errors section of the NewNet AccessManager Maintenance Manual.

Number of Parameters. (byte offset 12) – Specifies how many parameters are included in the report.

Length of Parameter 1 (byte offset 13) – Length of Parameter 1 in bytes.

Parameter 1 (byte offset 14) – Linkset number. Linkset numbers are derived by the order in which they appear in the MTP MML file, starting with 0.

Length of Parameter 2 (byte offset 15) – Length of Parameter 2 in bytes.

Parameter 2 (byte offset 16) – Link number (same as the SLC number).

6.7 ADVICE OF CHARGE (AOC) API DEFINITION

The Advice of Charge (AOC) feature allows a host AOC application to exchange charge-related information with AOC applications in other exchanges through Charging Tariff (CRGT), Charging Acknowledgement (CRGA), and Add On Charging (AOCRG) messages. These messages enable the exchanges belonging to different networks involved in the path of the call to understand the charging related information and use it for subscriber billing or AOC purposes.

To use the AOC feature in cktint:

- Feature Flag 14 must be enabled in the **CktInt.cfg** file.
- Debug Flag 41 may be enabled in the **CktInt.cfg** file to print AOC-related information in the log file.
- Variant must be set to "Generic" in ISUP Level Provisioning.
- If the exchange is a transit exchange for AOC, the following line must be added in the **CktInt.cfg** file.

  -AOCTRANSIT

  Otherwise, the exchange will be considered an originating or terminating exchange for AOC.
- The $49 Command parameter code must be specified as APP (Application Transport Parameter—0x78). This code will also be returned in subsequent $EA reports.

6.7.1 AOC Charging Message Types and Parameters

The AOC feature uses the charging messages listed below in the APP parameter:

Table 6.3: AOC Charging Messages

Hex Code	Message Type	Acronym
F1	Charging Tariff Information	CRGT
F2	Add On Charging Information	AOCRG
F3	Charging Acknowledgement Information	CRGA

The AOC feature uses the parameters listed below in the APP parameter:

Table 6.4: APP Parameters for AOC

Hex Code	Parameter Name
01	Extensions
02	Charging Control Indicators
03	Origination Identification
04	Destination Identification
05	Acknowledgement Indicators
06	Add On Charge Currency
07	Add On Charge Pulse
08	Current Tariff Control Indicators Currency
09	Next Tariff Control Indicators Currency
0A	Current Tariff Control Indicators Pulse
0B	Next Tariff Control Indicators Pulse
0C	Current Call Attempt Charge Currency
0D	Next Call Attempt Charge Currency
0E	Current Call Setup Charge Currency
0F	Next Call Setup Charge Currency
10	Current Call Attempt Charge Pulse
11	Next Call Attempt Charge Pulse
12	Current Call Setup Charge Pulse
13	Next Call Setup Charge Pulse
14	Currency
15	Current Communication Charge Sequence Currency
16	Next Communication Charge Sequence Currency
17	Current Communication Charge Sequence Pulse
18	Next Communication Charge Sequence Pulse
19	Tariff Switchover Time Currency
1A	Tariff Switchover Time Pulse

6.7.2 AOC Parameter Definitions

6.7.2.1 Extensions

The format for this parameter is shown below. This parameter is taken as a set of bytes. The validity of the parameter will not be checked by the call control.

	8	7	6	5	4	3	2	1
1	Extension							
.								
.								
99								

Extension – Extension data. This field can have a maximum of 99 bytes.

6.7.2.2 Charging Control Indicators

The format of this parameter is shown in the figure below:

	8	7	6	5	4	3	2	1
1	Number of bits in data							
2	Data							

Number of bits in data – Indicates the number of bits in the data field that are used, starting from the least significant bit. Since only three bits are valid, length should always be three.

Data – Contains the bit settings for Data. A maximum of eight bits can be set at a time. Only three bits are valid. Rest of the bits are spare. The first bit should be filled in the MSB (bit position 8), followed by the second bit. The bits should be filled as shown below:

0	1	2	x	x	x	x	x

0 = Subscriber Charge

1 = Immediate change of Actually Applied Tariff

2 = Delay Until Start

6.7.2.3 Origination Identification

The format of these parameters is shown in the figure below:

	8	7	6	5	4	3	2	1
1	Reference ID							LSB
2								
3								
4	MSB							
5	Length of National Regulation							
5a	National Regulation							
5b								
5c								
5d								
6	Length of Network							
6a	Network							
6b								
6c								
6d								
7	Length of Node Identification							
7a	Node Identification							
7b								
7c								
7d								

Reference ID – Integer value represents the Reference ID. It has the size of four bytes.

Length of National Regulation – Specifies the number of bytes in the National Regulation authority. Length cannot be more than 4 bytes.

National Regulation Authority – Value of National regulation authority. If n is the value of Length of National Regulation Authority field, then this field will have n bytes starting with the least significant byte.

Length of network field – Specifies the number of bytes in Network. Length cannot be more than 4 bytes.

Network – Value of network. If n is the value of Length of network field, then this field will have n bytes starting with the least significant byte.

Length of Node Identification field – Specifies the number of bytes in Node identification. Length cannot be more than 4 bytes.

Node identification – Value of node. If n is the value of Length of Node Identification field, then this field will have n bytes starting with the least significant byte.

6.7.2.4 Destination Identification

Same as Origination Identification.

6.7.2.5 Acknowledgement Indicators

The format of this parameter is shown in the figure below:

	8	7	6	5	4	3	2	1
1	Number of bits in Accepted							
2	Accepted data							

Number of bits in Accepted – Indicates the number of bits in the Accepted data field that are used starting from the least significant bit. Since only 1 bit is valid length should be 1.

Accepted data – Contains the coding for Accepted data. A maximum of 8 bits can be set at a time. Only one bit is valid. Rest of the bits is spare. The first bit should be filled in the MSB (bit position 8). The bit should be filled as shown below:

0	x	x	x	x	x	x	x

0 = Accepted

6.7.2.6 Add On Charge Currency

The format of this parameter is shown in the figure below:

	8	7	6	5	4	3	2	1
1								LSB
2				Currency Factor				
3	MSB							
4	Currency Scale							

Currency Factor – It is an integer value with a size of 3 bytes

Currency Scale – It is an integer value with a size of 1 byte.

6.7.2.7 Add On Charge Pulse

The format of this parameter is shown in the figure below:

	8	7	6	5	4	3	2	1
1	Pulse units							

6.7.2.8 Current Tariff Control Indicators Currency

The format of this parameter is shown in the figure below:

	8	7	6	5	4	3	2	1
1	Number of bits in Non Cyclic Tariff							
2	Non Cyclic Tariff							

Number of bits in Non Cyclic Tariff – Indicates the number of bits in the Non Cyclic Tariff field that are used starting from the least significant bit. Since only 1 bit is valid, length should be 1.

Non Cyclic Tariff – Contains the coding for Non-cyclic Tariff. A maximum of 8 bits can be set at a time. Only one bit is valid. Rest of the bits is spare. The first bit should be filled in the MSB (bit position 8). The bit should be filled as shown below:

0	x	x	x	x	x	x	x

0 = Non-Cyclic Tariff

6.7.2.9 Next Tariff Control Indicators Currency

Same as Current Tariff Control Indicators Currency.

6.7.2.10 Current Tariff Control Indicators Pulse

Same as Current Tariff Control Indicators Currency.

6.7.2.11 Next Tariff Control Indicators Pulse

Same as Current Tariff Control Indicators Currency.

6.7.2.12 Current Call Attempt Charge Currency

The format of this parameter is shown in the figure below:

	8	7	6	5	4	3	2	1
1	Currency Factor							LSB
2								
3	MSB							
4	Currency Scale							

Currency Factor – It is an integer value with a size of 3 bytes

Currency Scale – It is an integer value with a size of 1 byte.

6.7.2.13 Next Call Attempt Charge Currency

Same as Current Call Attempt Charge Currency.

6.7.2.14 Current Call Setup Charge Currency

Same as Current Call Attempt Charge Currency.

6.7.2.15 Next Call Setup Charge Currency

Same as Current Call Attempt Charge Currency.

6.7.2.16 Current Call Attempt Charge Pulse

The format of this parameter is shown in the figure below:

	8	7	6	5	4	3	2	1
1	Pulse units							

6.7.2.17 Next Call Attempt Charge Pulse

Same as Current Call Attempt Charge Pulse.

6.7.2.18 Current Call Setup Charge Pulse

Same as Current Call Attempt Charge Pulse.

6.7.2.19 Next Call Setup Charge Pulse

Same as Current Call Attempt Charge Pulse.

6.7.2.20 Currency

The format of this parameter is shown in the figure below. **The currency should be a hex value in the $49 command.**

	8	7	6	5	4	3	2	1
1	Currency							

6.7.2.21 Current Communication Charge Sequence Currency

The format of this parameter is shown in the figure below. A maximum of four sets of Currency Factor, Currency Scale, Tariff Duration, Number of bits in Sub Tariff Control, Sub Tariff Control data fields can be present in this parameter in the sequence shown in the figure below:

	8	7	6	5	4	3	2	1
1a								LSB
1b				Currency Factor				
1c	MSB							
1d				Currency Scale				
1e				Tariff Duration				LSB
1f	MSB							
1g				Number of bits in Sub Tariff Control				
1h				Sub Tariff Control Data				
2a								
.								
.								
4a								LSB
4b				Currency Factor				
4c	MSB							
4d				Currency Scale				
4e				Tariff Duration				LSB
4f	MSB							
4g				Number of bits in Sub Tariff Control				
4h				Sub Tariff Control Data				

Currency Factor – It is an integer value with a size of 3 bytes

Currency Scale – It is an integer value with a size of 1 byte.

Tariff Duration – It is an integer value with a size of 2 bytes.

Number of bits in Sub Tariff Control – Indicates the number of bits in the Sub Tariff Control Data field that are used starting from the least significant bit. Since only one bit is valid, length should be 1.

Sub Tariff Control Data – Contains the coding for Sub Tariff Control Data. A maximum of 8 bits can be set at a time. Only one bit is valid. Rest of the bits is spare. The first bit should be filled in the MSB (bit position 8). The bit should be filled as shown below:

0	x	x	x	x	x	x	x

0 = One Time Charge

6.7.2.22 Next Communication Charge Sequence Currency

Same as Current Communication Charge Sequence Currency.

6.7.2.23 Current Communication Charge Sequence Pulse

The format of this parameter is shown in the figure below. The Pulse Units, Charge Unit Time Interval, Tariff Duration fields can be present in this parameter for a maximum of four times in the sequence shown in the figure below:

	8	7	6	5	4	3	2	1
1a	Pulse Units							
1b	Charge Unit Time Interval							LSB
1c	MSB							
1d	Tariff Duration							LSB
1e	MSB							
.								
.								
.								
4a	Pulse Units							
4b	Charge Unit Time Interval							LSB
4c	MSB							
4d	Tariff Duration							LSB
4e	MSB							

Pulse Units – It is an Octet String with a size of 1 byte.

Charge Unit Time Interval – Tariff Duration:

These are integer values with a size of two bytes each.

6.7.2.24 Next Communication Charge Sequence Pulse

Same as Current Communication Charge Sequence Pulse.

6.7.2.25 Tariff Switchover Time Currency

The format of this parameter is shown in the figure below:

	8	7	6	5	4	3	2	1
1	Tariff Switchover Time							

6.7.2.26 Tariff Switchover Time Pulse

Same as Tariff Switchover Time Currency.

6.7.3 AOC Messages and Parameters

The following tables specify which parameters are allowed in each message type and whether the parameters are optional or mandatory.

Table 6.5: Charging Tariff Information (CRGT) Message

Parameter Name	Type
Charging Control Indicators	Mandatory
Currency	Mandatory
Current Tariff Control Indicators Currency[1]	Mandatory
Current Tariff Control Indicators Pulse[2]	Mandatory
Next Tariff Control Indicators Currency[1]	Mandatory
Next Tariff Control Indicators Pulse[2]	Mandatory
Origination Identification	Mandatory
Tariff Switchover Time Currency[1]	Mandatory
Tariff Switchover Time Pulse[2]	Mandatory
Current Call Attempt Charge Currency[1]	Optional
Current Call Attempt Charge Pulse[2]	Optional
Current Call Setup Charge Currency[1]	Optional
Current Call Setup Charge Pulse[2]	Optional
Current Communication Charge Sequence Currency[1]	Optional
Current Communication Charge Sequence Pulse[2]	Optional
Destination Identification	Optional
Extensions	Optional
Next Call Attempt Charge Currency[1]	Optional
Next Call Attempt Charge Pulse[2]	Optional
Next Call Setup Charge Currency[1]	Optional
Next Call Setup Charge Pulse[2]	Optional
Next Communication Charge Sequence Currency[1]	Optional
Next Communication Charge Sequence Pulse[2]	Optional

1. Tariff Currency-related parameter
2. Tariff Pulse-related parameter

NOTE: Only one of the sets of parameters, Tariff Currency or Tariff Pulse, can be present in a single CRGT message.

The Charging Tariff information in a CRGT message can be represented as either a Tariff Currency or Tariff Pulse. If it's represented in Tariff Currency format, the data is divided into two parts: Current Tariff Currency and Tariff Switch Currency. Tariff Currency formats can have either one or both of these parts at the same time. If Current Tariff Currency is present, then the Current Tariff Control Indicator Currency parameter is mandatory. If the Tariff Switch Currency parameter is included, then the Next Tariff Control Indicator Currency and Tariff Switchover Time Currency parameters are also mandatory.

When the Charging Tariff information is represented as a Tariff Pulse format, the data is divided into two parts: Current Tariff Pulse and Tariff Switch Pulse. Tariff Pulse formats can have either one or both of these parts at the same time. If Current Tariff Pulse is present, then the Current Tariff Control Indicator Pulse parameter is mandatory. If the Tariff Switch Pulse is included, then the Next Tariff Control Indicator Pulse and Tariff Switchover Time Pulse parameters are also mandatory.

Table 6.6: Add On Charging Information (AOCRG) Message

Parameter Name	Type
Add On Charge Currency[1]	Mandatory
Add On Charge Pulse[2]	Mandatory
Charging Control Indicators	Mandatory
Currency	Mandatory
Origination Identification	Mandatory
Destination Identification	Optional
Extensions	Optional

1. Charge Currency-related parameter
2. Charge Pulse-related parameter

NOTE: Only one of the parameters, Charge Currency or Charge Pulse, can be present in a single AOCRG message.

Table 6.7: Charging Acknowledgement Information (CRGA) Message

Parameter Name	Type
Acknowledgement Indicators	Mandatory
Destination Identification	Mandatory
Origination Identification	Mandatory
Extensions	Optional

6.7.4 Application Transport Parameter (APP) Coding

6.7.4.1 APP Format for ITU V5.2 AOC Feature

The charging data can be carried in APM or PRI ISUP messages. All the data values of charge parameters should be coded in hex format. If the ACI is other than 0x83, or even if the ACI is 0x83 and the exchange is a “pass on” exchange for AOC, then the whole information after the “length” field will be transparently passed to the network. Cktint will not validate the information in the APP parameter.

Similarly, when a message is received from the network, if the ACI is other than 0x83, or if the ACI is 0x83 and the exchange is a “pass on” exchange for AOC, the APP parameter will be transparently passed to the host in an $EA report without any BER decoding.

The parameter segment for the APP parameter in the $49 command and $EA report of ISUP messages will have the following format:

Name Length ACI ATII SI SLR HCSI CNSI MessageType Parameter Segments

Name – APP parameter code—0x78

Length – Specifies how many octets of data follow “length” in the APP parameter. Convert from decimal to hex.

> *NOTE: The following APP field formats are based on ITU Recommendation Q.763 Addendum 1 (05/98)*

Application Context ID (ACI) – Specifies the Application Context ID. The format for ACI is shown below.

8	7	6	5	4	3	2	1
Extension	Application Context Identifier						

The extension bit is always coded as 1.

For AOC, the ACI value is 3 and has a value of 0x83 in the API.

Application Transport Instruction Indicator (ATII) – Specifies the Application Transport Instruction Indicator field of the APP parameter. The format for ATII is shown below:

8	7	6	5	4	3	2	1
Extension	Spare					ATII	
						B	A

The extension bit is always coded as 1.

Only the least significant two bit fields are valid. These fields can have the following possible values:

Bit A – Release call indicator

0 = do not release call

1 = release call

Bit B – Send notification indicator

0 = do not send notification

1 = send notification

Segmentation Indicator (SI) – This value indicates the APM Segmentation Indicator in the APP parameter received from the network. The format for Segmentation Indicator (SI) is the same as that of the APM Segmentation Indicator and Sequence Indicator shown below and has the same meaning in an $EA report.

8	7	6	5	4	3	2	1
Extension	SI	APM Segmentation Indicator					

The extension bit is always coded as 1.

The APM Segmentation Indicator (bits 1-6) can have the following possible values:

0 = final segment

1-9 = indicates the number of following segments

10-255 = spare

The Sequence Indicator (SI-bit 7) can have the following possible values:

0 = subsequent segment to first segment

1 = new sequence

In a $49 command this has a different meaning. Due to the size limitation when the host is segmenting the APP parameter, the size of the complete charging data from the host is filled in the SI field. If the size of the charging data is greater than 255 bytes, then the bits FED of HCSI are used as multiplication factors in combination with the value set in the SI so that the size of the complete charging data is sent in the $49 command.

Segmentation Local Reference (SLR) – The host application should always set this value if it's a terminating exchange for ACI (0x83), even if it does not segment the message. The format for SLR is the same as the Segmentation Local Reference (SLR) shown below. Cktint will further use this value if the message gets segmented during BER encoding.

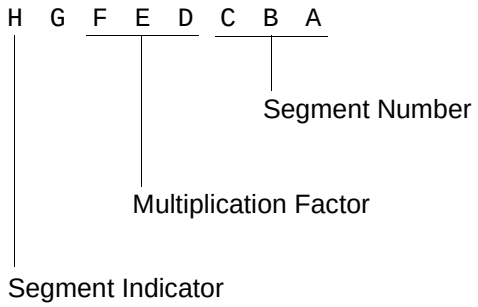
8	7	6	5	4	3	2	1
Extension	SLR						

The extension bit is always coded as 1.

NOTE: The following fields will all be BER encoded and together will form the Encapsulated Application Information.

Host/Cktint Segmentation Indicator (H/C SI) – This field is used to indicate whether the host application segmented the $49 command due to the API size limitation. Similarly, this field is used to indicate whether Cktint segmented the incoming AOC message from the network in the $EA report due to API size limitation.

The format of this field is shown below:



Segment Number (A B C) – Indicates the number of the segment. A maximum of 7 segments can be sent.

000 = No segmentation

001 = First segment

.

.

.

111 = Seventh segment

Multiplication Factor (D E F) – This field is used in combination with the SI value to determine the size of the charging data from the host application. The bits D, E, and F are set in the $49 command when the host is sending a segmented APP parameter to give information about the size of the complete charging data.

Example:

If the host is sending charging data whose size is 510 bytes:

The SI value can be set to 0xFF.

The multiplication factor (bits DEF of H/C SI) can be 010 (0x02).

Thus, the size of the APP parameter from the host is taken as 255 * 2 = 510 bytes.

If the host is sending charging data whose size is 511 bytes, which is an odd number, the closest approximation of the value is set:

The SI value can be set to 0xAB

The multiplication factor (bits DEF of H/C SI) can be 011 (0x03).

Thus, the size of the APP parameter from the host is taken as 171 * 3 = 513 bytes.

Segment Indicator (H) – Indicates whether it's a final segment or not.

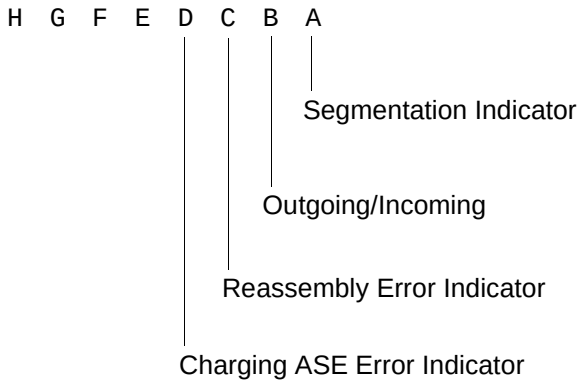
1 = Final Segment

0 = Not final segment

The Host/Cktint Segmentation indicator should be 00 if the host application does not segment the $49 command for AOC.

Cktint/Network Segmentation Indicator (C/N SI) – This field indicates whether Cktint segmented the AOC message from the host during BER encoding or the message received from the network is segmented. Cktint will set this field in the $EA report. It does not have any meaning in the $49 command. It should be set to 00 in a $49 command.

The format of this field is shown below:



Segmentation Indicator (A) – Indicates whether the message is segmented or not.

1= segmentation

0 = no segmentation

Outgoing/Incoming (B) – Indicates whether the message is segmented by Cktint during BER encoding of an outgoing message or whether the message received from the network is segmented.

1 = Outgoing

0 = Incoming

Reassembly Error Indicator (C) – During reassembly, if an error occurs in Cktint, this field will be set in the $EA report of an APM message to inform the host.

0 = No error

1 = Error

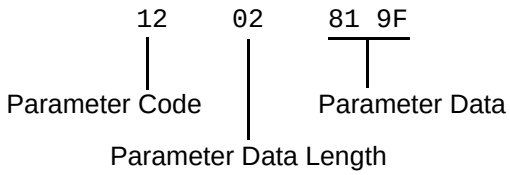
Charging ASE Error Indicator (D) – If an error occurs during decoding of an incoming APP parameter by Charging ASE, this field will be set in the $EA report of an APM message to inform the host. The host should take appropriate action.

0 = No error

1 = Error

Message Type – Specifies the type of AOC message. (CRGT or AOCRG or CRGA)

Parameter Segments – This field should have the following format:



Parameter Code – Specifies the parameter code.

Parameter Data Length – Specifies the number of parameter data bytes (in hex).

Parameter Data – Specifies the parameter data (in hex).

Example:

Assume the following information data of CRGT needs to be sent from host to cktint in a single $49 command.

Application Context Id	83
Application Transport Instruction Indicator	82
Segmentation Indicator (SI)	00
Segmentation Local Reference (SLR)	20
Host Cktint Segmentation Indicator (HCSI)	00

Cktint Network Segmentation Indicator (CNSI)	00
Message Type	F1
Charging Control Indicator	03 E0
Origination Identification	0A 02 03 06 32 07 02 02 05 02 07 0A
Currency	11
Current Communication Charge Sequence Currency	22 01 03 02 2E 00 01 80
Current Tariff Control Indicators Currency	01 80
Next Tariff Control Indicators Currency	01 80
Tariff Switchover Time Currency	04
Current Call Attempt Charge Currency	02 3E 00 01

Current Call Setup Charge Currency	05
	6A
	00
	01
Next Communication Charge Sequence Currency	02
	00
	03
	02
	2E
	00
	01
	80
Next Call Attempt Charge Currency	01
	2E
	00
	01

The coding of an APP parameter with CRGT message information (as defined above) in the $49 command is shown below:

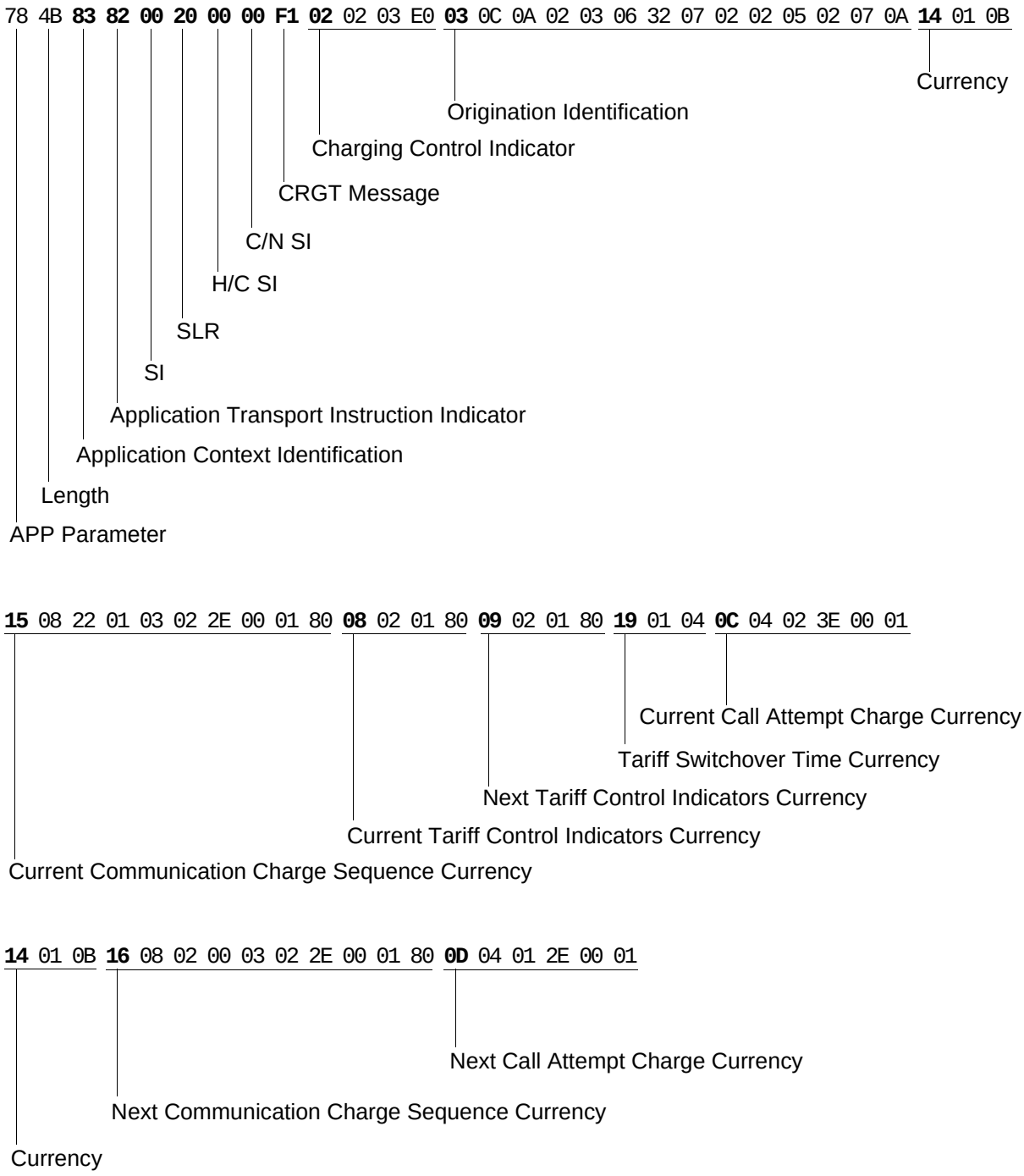


Figure 6.9: Example APP Parameter Coding with CRGT Message Information in $49 Command

6.7.5 AOC Usage Guidelines for Host Application

The following functionality is implemented to handle the Advice of Charge (AOC) feature in the SS7 subsystem.

6.7.5.1 Outgoing Messages

When the exchange is an originating exchange for ACI 0x83 (AOC):

- As per the API definition for AOC, the charging messages —CRGT, AOCRG and CRGA—have both mandatory and optional parameters. Validations are performed by Cktint so that a $49 command from the host containing a Charging message without one or more of the mandatory parameters will not be allowed.

  If the host sends a $49 command with a charging message that has one or more mandatory parameters missing, then that $49 command is rejected with NSB 0xCB (NSB_PARAM_ERROR_IN_SEGMENT).

- As per the protocol, an AOCRG request primitive and CRGA request primitive is allowed after start of charging only. The host application should ensure that either AOCRG or CRGA is not transmitted before the transmission or reception of a CRGT message. If the SS7_message is an IAM, and the charging message type is AOCRG or CRGA, then the $49 command is rejected with a NSB 0xCF (NSB_PROTOCOL_VIOLATION).

- If the charging data cannot fit in a single $49 command of an ISUP message like ACM, CPG, CON, ANM, or PRI, the host application can segment the charging data and send the subsequent segments only in APM messages.

  – In the first segment, bits CBA in H/C SI should be set to 001 and bit H should be zero to indicate that it's a first segment.

  – In the subsequent segments bits, CBA should be incremented to indicate the corresponding segment number.

  – Bit H should be set to 1 in the final segment.

  – The size of the complete charging data is filled in the SI field. If the size of the charging data is greater than 255 bytes, then bits FED of H/C SI are used as a multiplication factor in combination with the value set in the SI so the size of the complete charging data is sent in the $49 command.

  – In all the segments, the SLR value should be the same. If the SLR value differs, the $49 command will be rejected with NSB_INVALID_SLR (0x D5) and the previous segments will be discarded.

  If the complete charging data can fit in a single $49 command, due to the API size limitation, the host application should still set the SLR value in the $49 command so that even if the data gets segmented during BER encoding in the call control, this SLR value will be used.

  NOTE: If the charging data cannot fit in a single $49 command and the host is segmenting the data, then the host application should send the message type of the charging message in each segment.

- There's always a possibility that the application data could be lost if sent during call setup (i.e. Sent in an IAM before the receipt of a backward message.). To avoid this, if the host application has to send the charging data in an IAM message:
  - The $49 command should always contain an APP parameter with empty data.
  - The ACI should be set to 0x83, ATII to whatever is required by the host, SLR to a unique value from 0 to 126, and HCSI to 01(not final segment, but first segment).
  - Once an acknowledgment is received from the network, the host can send the rest of the segments in subsequent APM messages with the same SLR value as the previous segment (in previous IAM) and H/C SI indicating the appropriate segment number.
  - When the host sends the final segment, bit H in H/C SI should be set to 1.

  In other words, if the host application wishes to send the charging data in the IAM message, the message will always be segmented. The host application should make sure that it does not send the application data before a backward message is received from the network
- If the SS7 message is an IAM with a CRGT charging message, then the length of the APP parameter is checked. If the length is greater than 0x7, then the $49 command is rejected with NSB 0xCB (NSB_PARAM_ERROR_IN_SEGMENT).

  If the host application wants to send Charging data in an IAM message, the $49 command for an IAM should always contain an APP parameter with empty data. The host application can send the real application data in subsequent APM messages after a backward message has been received.
- If Cktint segments the application data from the host while BER encoding due to the MTP limitation of 272 octets:
  - The ISUP message from the host will be sent to the network with an APP parameter with “0” charging data and SI in the set to “New Sequence.”
  - Cktint will send rest of the segments in separate APM message to the network with the SLR value set to the one sent from the host for that ISUP message.
  - For each of the APM messages generated, an $EA report with message type as APM and an APP parameter segment will be sent to the host to indicate that the message from the host has been segmented by Cktint.
  - The SI is set for each of the $EA reports with the same value as that sent to the network.
  - Bit A in the C/N SI field will be set to 1 to indicate that segmentation has occurred and bit B will be set to 1 to indicate outgoing segmentation. Bits A and B together indicate that Cktint segmented the outgoing message.
  - The message type field will be set to the corresponding AOC message type.
  - ACI, ATII and SLR fields in the $EA report will be set to the one sent by the host in the $49 command for that ISUP message.
  - The SI field will be zero.
- If Cktint does not segment the ISUP message during BER encoding, the complete charging data will be sent in a single ISUP message. SLR values from the host will be dropped.

- If the charging data does not fit in a single $49 command of a PRI message, host application can send the rest of the data in APM messages, but it should also make sure that a REL message is sent after that. Cktint will not validate the message sent after PRI.
- When the host application segments the charging data it should make sure that subsequent segments have the same SLR value and the H/C SI fields CBA are incremented in subsequent segments. If there's a mismatch in the H/C SI field with the previous $49 command, the current $49 command will be rejected with NSB_INVALID_HOST_CKTINT_SI (0x D6).
- While processing the segmented $49 commands from the host, if a REL message is received from the host application, the whole charging data will be discarded and not sent to the network.

When the exchange is a pass-on exchange for ACI 3 or any other ACI:

- The application data in the APP parameter will be transparently passed to the network. BER encoding of the application data will not be done by Cktint. The validity of the fields after “length” in the APP parameter of the $49 command will not be checked. In other words, Cktint will assume that the information after “length” field will be in the same format as specified in Recommendation Q.763 Addendum 1.

6.7.5.2 Incoming Messages

When the exchange is a terminating exchange for ACI 0x83 (AOC):

- When an IAM or any other ISUP message is received from the network with ACI 3, and there's no segmentation:
  - The whole charging information will be passed to the host after BER decoding in the $EA report for that ISUP message.
  - The SI field will be the same as the one received from the network.
  - SLR, H/C SI, and C/N SI will be set to zero.
- When an IAM message is received from the network with ACI 3, and if there's segmentation:
  - An $EA report for this IAM will be sent to the host with “0” data in the APP parameter.
  - Bit A of C/N SI will be set to 1 and bit B will be set to 0. Bit A and B together indicates that a segmented message has been received from the network.
  - SLR, ACI, ATII and SI values will be set to the one present in the APP parameter received from the network.
  - H/C SI will be set to zero.
  - The host application should respond with a backward message with empty APP parameter data and the same SLR value as received from the network. It should also start the timer Treass.

- If subsequent segments are received in APM messages:
  - *An $EA report will be sent to the host for each of the APM messages received, but the application data will be empty.*
  - *The SI value will be set to the one received from the network in the APP parameter.*
  - *C/N SI will be set to indicate segmentation from the network.*
  - *The ACI, ATII, and SLR values will be the same as the one received from the network.*
  - *H/C SI will be zero.*
- When the APP parameter is received from the network during Charging ASE processing, and when a Charging ASE error occurs, then the bit D of C/N SI is set to 1 to indicate Charging ASE error has occurred. The host should take appropriate action in this situation. At all other times, this bit is set to zero.
- After complete reassembly, the whole information will be sent to the host in the $EA report of an APM message. The ACI, ATII, and SLR values will be the same as the one received from the network. The SI field will have the same value as in the APP parameter of the final segment received from the network.
- If the reassembled application data is longer than the host API size is, the information will be segmented and sent in subsequent $EA reports for APM messages. Bits CBA in H/C SI will represent the segment number and bit H will indicate whether it's a final segment or not. In all the $EA reports, the SI value will be that of the final segment received from the network
- If an error occurs during reassembly, Cktint will send an $EA report to the host application for APM message. Bit C in C/N SI will be set to indicate reassembly error. The host application should stop the timer Treass and take appropriate action.
- If a REL message is received from the network during reassembly, Cktint will stop the reassembly procedure and discard the previous segments.
- If the timer Treass fires during reassembly, the host application should take appropriate action based on the ATII value set in the received APP parameter from the network
- The host application should also handle Unidentified Context Error.

When the exchange is a pass-on exchange for ACI 0x83 or any other ACI:

- If an IAM or any ISUP message is received from the network, an $EA report for this message will be sent to the host. The application data in the APP parameter will be transparently passed to the host without any BER decoding. In other words, Cktint will assume that the format of the received APP parameter is same as specified in Recommendation Q.763 Addendum 1. Cktint will fill the “name” and “length” field for APP parameter in $EA report. The rest of the data will be the same as received from the network.

6.7.5.3 Host Application Special Requirement

The following functionality should be handled by the host application:

- Reassembly timer Treass and Tcrga should be handled by the host application.
- Unidentified context error, Reassembly error, and Charging ASE error should be handled by the host application.
- If the host application wants to send Charging data in an IAM message, the $49 command for an IAM should always contain an APP parameter with empty data. The host application can send the real application data in subsequent APM messages after a backward message has been received.
- An AOCRG request primitive and CRGA request primitive are allowed after start of charging only. The host should ensure that AOCRG and CRGA messages are not allowed before the start of charging (before a CRGT message).

6.7.6 Timers

Handling of Timer Tcrga and Treass is the responsibility of the host application. The host application should set the timer and take appropriate action on timer expiration.

6.8 CALL FLOW EXAMPLES

This section provides you with three call flow examples. Each example shows the interaction between the various components on the network.

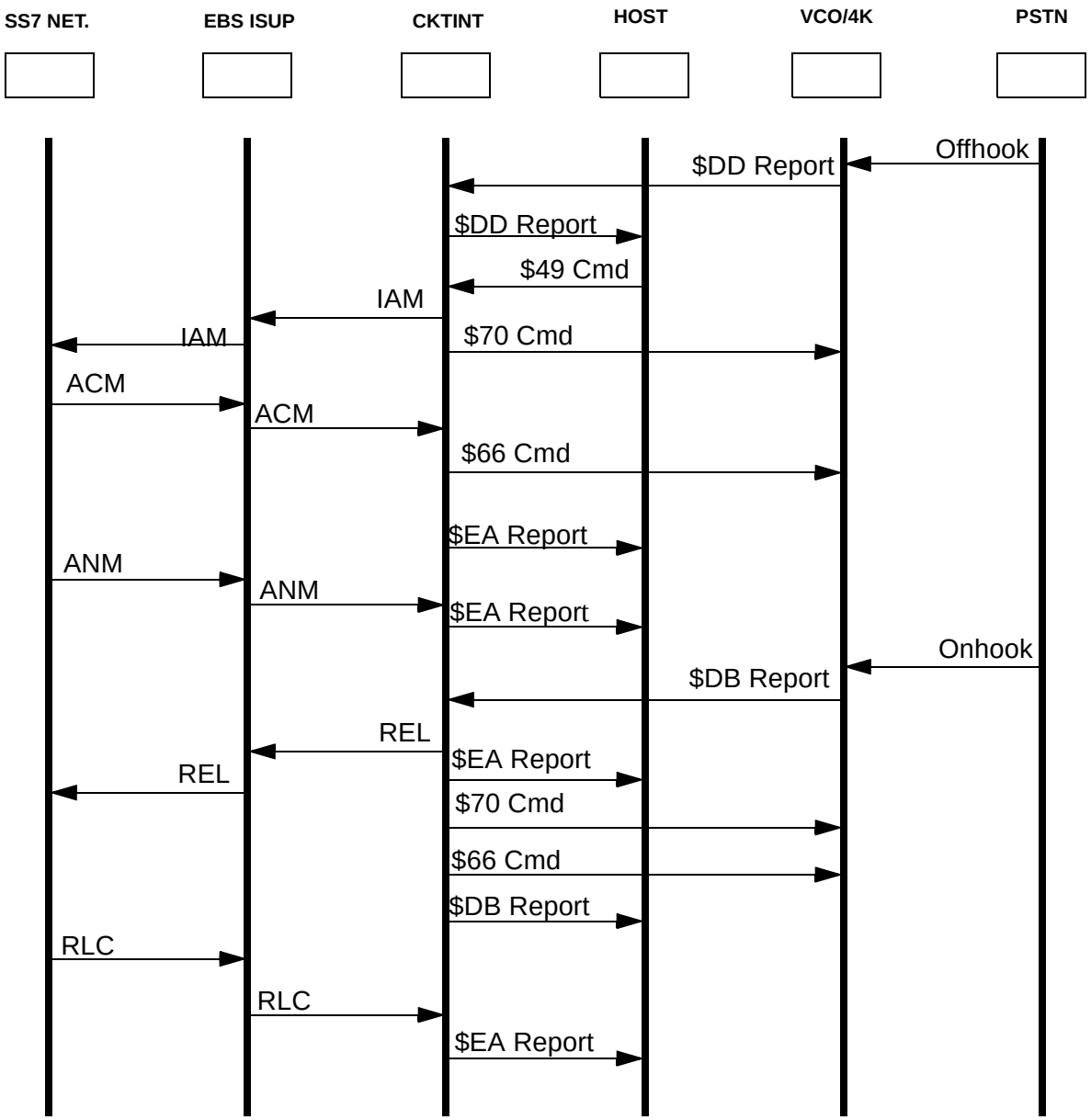


Figure 6.10: Non-SS7-To-SS7 Call Establishment

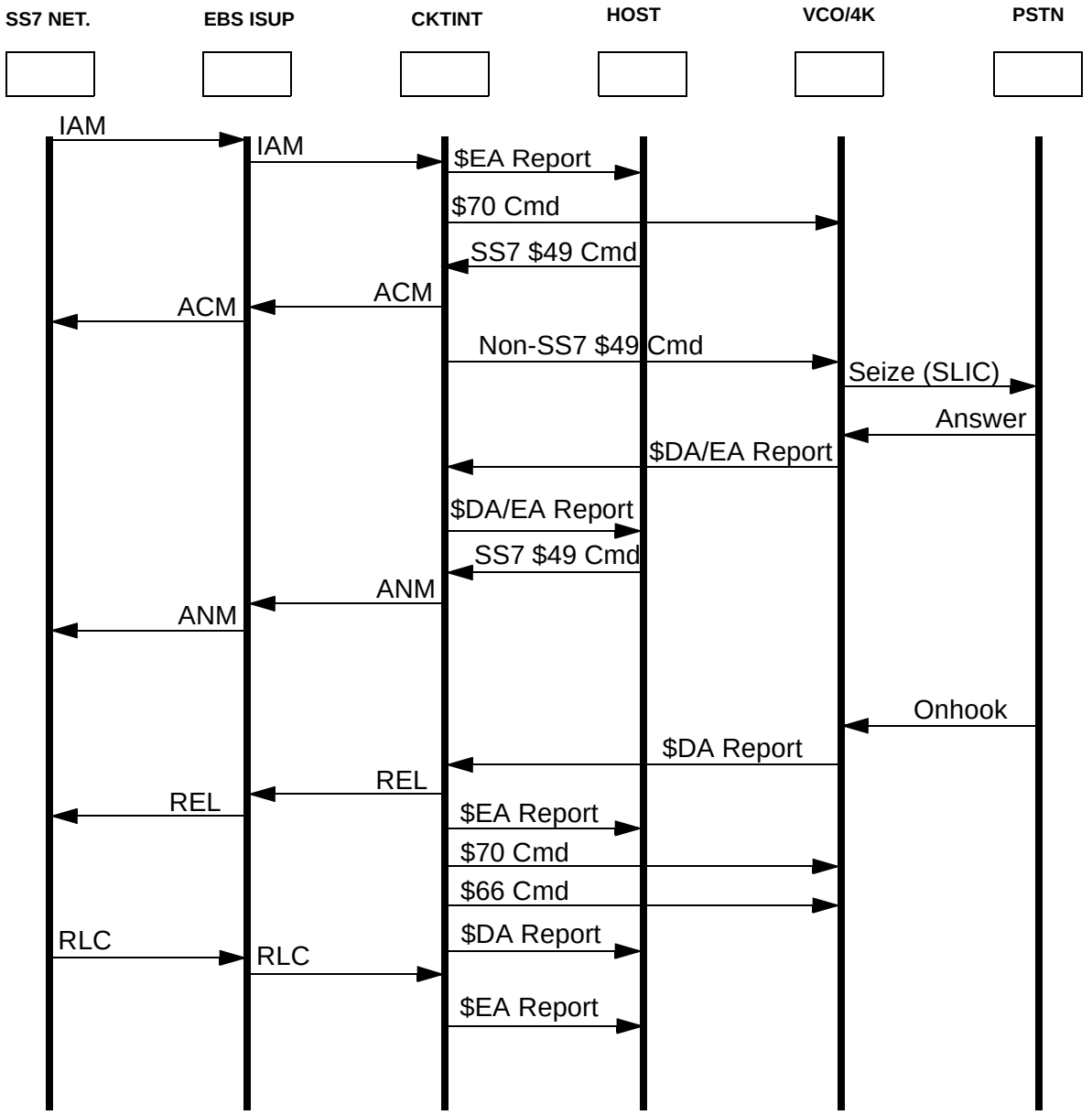


Figure 6.11: SS7-to-Non-SS7 Call Establishment

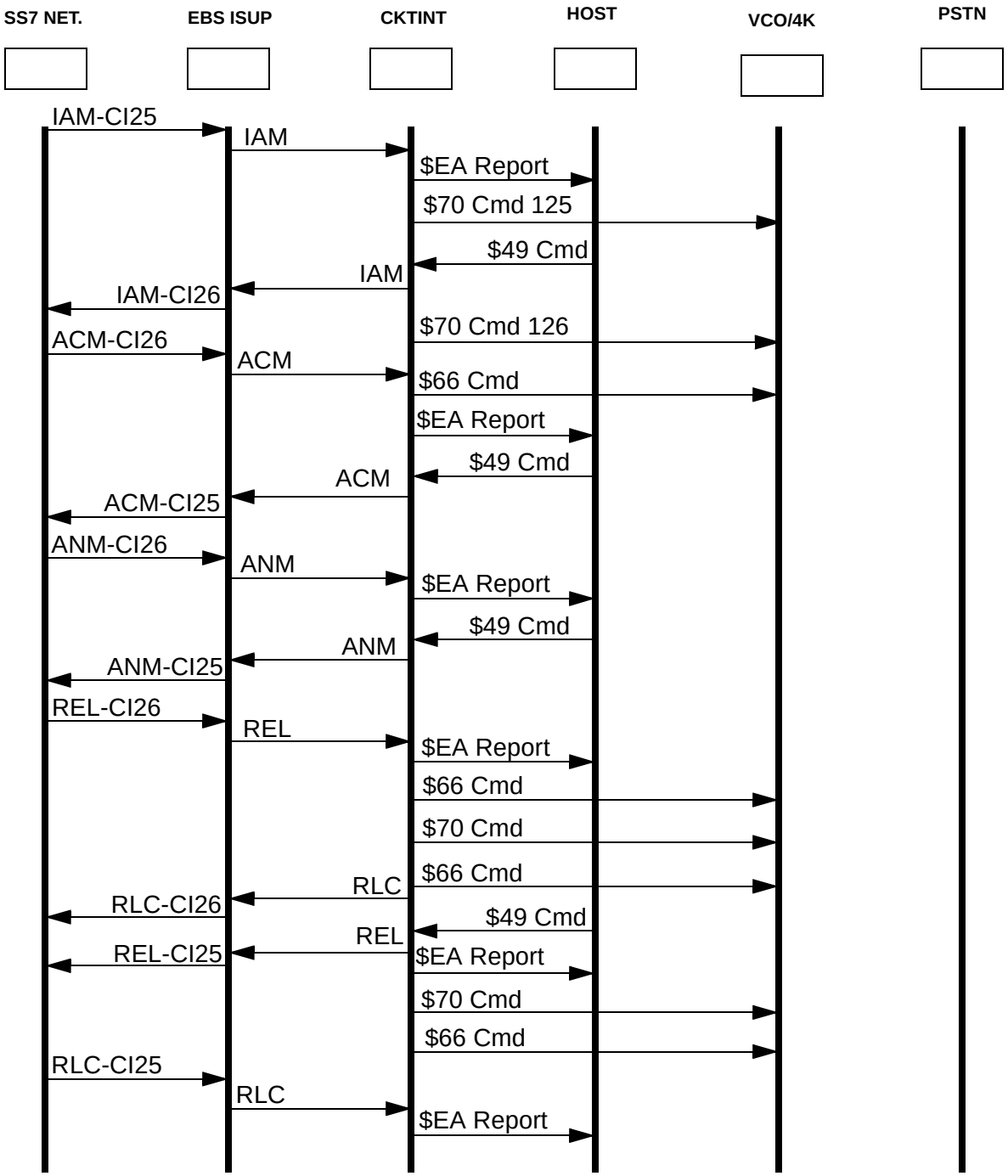


Figure 6.12: SS7-to-SS7 Call Establishment

6.8.1 Outbound COT Call Flow Examples

If your system is configured for outbound COT, the following events occur when the host sends an IAM with "continuity check required on this circuit."

1. CktInt sends the IAM to the SS7 stack.
2. The SS7 stack responds to CktInt with a primitive type of ISUP_START_OGCCOT, and CktInt puts the circuit's substate to CONT_INITIATED.
3. CktInt sends a $66 command to the VCO/4K to attach an ISUP Tone (0x4D9 or 0x4D8) to the specific voice port. CktInt also sends a $69 command to perform the outpulse rule.
4. The SS7 stack sends primitive type ISUP_START_CHECKTONE to CktInt.

At this point, one of two sets of events occur:

1. CktInt receives a $DA report from the VCO/4K indicating that an ISUP Tone was detected (0x3107):
2. The circuit's substate goes to CONT_INITIATED_DETECTED.
3. CktInt responds to the SS7 stack with ISUP_BWD_CHECKTONE.
4. The SS7 stack sends CktInt an ISUP_STOP_CHECKTONE.
5. The circuit's substate goes to INITIATED.
6. CktInt sends ISUP_STOP_CHECKTONE to the SS7 stack and the stack sends a COT (success) to the network.
7. The SS7 stack sends Cktint an ISUP_STOP_OGCCOT.
8. CktInt sends a $66 command to the VCO/4K to make the port listen to quiet (0x4C0).
9. The call proceeds with the network responding with an ACM.

—OR—

1. CktInt receives ISUP_STOP_CHECKTONE.
2. The circuit's substate goes to CONT_INITIATED_END.
3. CktInt receives a COT with primitive type ISUP_STOP_OGCCOT, indicating a COT failure.
4. CktInt receives a release with primitive type ISUP_REATTEMPT.
5. CktInt sends the COT (failure) to the host.
6. CktInt receives an ISUP_START_OGCCOT with CCR from the network and sends it to the host.

   NOTE: The host needs to re-initiate another outbound IAM upon a continuity failure.

Section 7
TCAP

7.1 INTRODUCTION

TCAP software consists of Signaling End Point Translator (SEPT) and SS7 stacks. See *Section 3.2* for instructions on installing TCAP software and *Section 3.3* for a software configuraton checklist.

SEPT provides the following capabilities:

- Connectivity — Allows signaling systems residing in SS7 and TCP/IP domains the ability to freely converse with one another.
- Mapping — Resolves differences relating to addressing techniques and resource allocation used within SS7 and TCP/IP domains.
- Transformation — Resolves differences relating to signaling definition and content used within SS7 and TCP/IP domains.
- Network Management — Presents and encapsulates the functions of connectivity, mapping, and transformation to the end user in such a way that the complexities of the networking environments are removed or greatly reduced.

SEPT resides as an endpoint on the SS7 network, acting as a gateway to other networks of differing protocols. SEPT terminates an SS7 linkset on the endpoint of an SS7 network, as well as terminating TCP/IP. SEPT provides the following functionality:

- Protocol Bridge — Allows signaling to cross to and from an SS7 domain and a TCP/IP domain.
- Active signaling component — SEPT has the ability to modify and analyze signaling between an SS7 controlling device and TCP/IP domain.

Section 7.5 contains examples that show how to interface SEPT as TCP/IP client.

7.1.1 SEPT Components

The SEPT product consists of three major components:

- Service Execution Environment (SEE)/SS7 — A software module used to transmit and receive SS7 messages to and from data representations to system level calls at the TCAP/SCCP SS7 protocol layer.
- SEE/Transmission Control Protocol (TCP) — A software module used to transmit and receive TCP messages to and from data representations to system calls at the socket-level interface. (A socket-level interface is an Ethernet communication path.)
- Service Control Logic (SCL) — A software module used to actively manipulate the signaling stream.

Service Execution Environment (SEE)

The SEE is a software subsystem that provides the interface between SS7 protocol stacks (SEE/SS7), Ethernet stacks (SEE/TCP), and the applications running on top of them. SEE allows for a consistent presentation of messages to applications, regardless of which transport or protocol mechanism is being used. SEE also allows for the graceful turn-on and shut-down of applications running on the remote platforms. The SEE functionality includes:

- Protocol-independent platform for applications.
- Utilization of SS7 Signaling Connection Control Part/Transaction Capabilities Application Part (SCCP/TCAP) subsystem number (SSN) addressing mechanism.
- Vendor protocol independence.
- Simplification of application design and deployment of SCL.

Service Creation Logic (SCL)

The SCL allows manipulation of the signaling stream as it moves between the SEE/SS7 and SEE/TCP modules. The SCL application sits on top of the system software stack, which is dedicated to making decisions and controlling resources based on input from the different protocol streams.

The current SCL component simply passes signaling from one domain to the other. Manipulation of address destination and/or message content is not performed.

7.1.2 Multiple SEPT Sessions

The SEPT process can be invoked multiple times using different or equal subsystem number (SSN) address registrations. If the SSNs are equal, multiple SEPT process load share the same bi-directional SS7/TCAP traffic. The load sharing method provides process redundancy because other SEPT processes can continue to operate using the same SSN if one of the processes fail.

If the SEPT processes use different SSN addressing registrations, each SEPT process can serve different applications that are running simultaneously on the same physical platform.

7.2 SS7 SUBSYSTEM MESSAGE STRUCTURE COMPONENTS

The SS7 subsystem message structure is made up of two main components: a message header and a “protocol flavored” message. These components are shown in Figure 7.1.

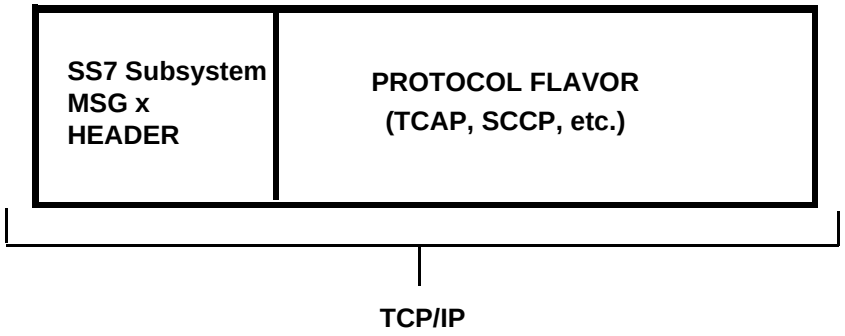


Figure 7.1: SS7 Subsystem Message Structure

The header contains identifiers for the protocol flavored message. The header also contains all addressing information needed for the protocol domain for which the signal is destined.

In the case of TCP/IP, the message header does not need to include the address information for the host side because virtual connections are made by SEPT and the TCP client at the time of service initiation. If acting as a TCP server, SEPT does not allow the ability to accept and release from multiple TCP/IP clients on a one-to-one basis.

7.2.1 Protocol Flavors

One of the principle functions SEE performs is the extraction and depositing of data structures into and out of protocol stacks that represent functions carried out by these protocols. As an example, with TCAP, SEE obtains all relevant data from the TCAP layer pertaining to an incoming signal, formats it into a defined structure, and transports it to the upper layers. This format or data structure representing an SS7 protocol layer transported over TCP/IP is referred to as a message flavor. When the SS7 subsystem transports TCAP information in this manner, it is transporting a TCAP flavor.

This concept of protocol flavor extends to other layers found within the SS7 protocol group. TCAP and SCCP are the supported protocol flavors.The essence of a protocol flavor is that it is a means of expressing operations within a layer of a protocol through the use of an encoded data structure. Any operation or result needed from a protocol layer can be initiated by generating or reading the contents of a flavored message packet.

Through the use of flavored messages, systems remote to actual SS7 protocol stacks can initiate operations on those stacks without the need to make direct system level calls. The SEPT process carries out these operations for them.

The format of these protocol flavors are based on C data structures that can be easily worked upon at the application layer. As a result, all services for the TCAP protocol definition can be realized through the manipulation of the SS7 subsystem signaling data structures.

Protocol flavors are the key to allowing non-SS7 based equipment access to SS7 services without special configuration or additional kernel/hardware modifications. If a node on an TCP/IP network can communicate with the SS7 subsystem over TCP/IP, the node has the ability to access and use SS7 services.

7.2.2 Generic SS7 Subsystem Header File

The following is the data structure for a generic SS7 subsystem header.

```
typedef struct IuMsgX
{   DWORD   HdrId;
    DWORD   SessionId;
    DWORD   status;
    DWORD   state;

    DWORD   Ss7AdrFlag;
    DWORD   Ss7AdrType;
    IUADR   Ss7Adr;
    DWORD   InetAdrFlag;
    DWORD   InetAdrType;
    IUADR   InetAdr;

    DWORD   MsgFlavor;
    DWORD   MsgLen;
    BYTE    msg[1];
} IUMSGX;
```

The following is a detailed description of the header.

HdrId — An identifier specifying the master message format must be set to GENERIC_SS7IU_HDR.

SessionId — This is the Session ID. This is a user defined field that would generally be used when coordinating activities between a remote host application and a custom SCL. Valid range is 0 to 4, 294, 295, 967.

status — All messages flowing from SEPT to a network host will have the status field set to true or false. Valid signals are accompanied with a TRUE indicator in the field. When tagged with a FALSE indication, it indicates a failure of a preceding signaling operation sent from a remote host to SEPT. This failed signal is simply a reflection back of the original signal sent by the host.

state — This is a command flag set by the host application which allows the suspension of SEPT/SEE services as seen by remote SS7 signaling points. Set to ACTIVE to put in service when sending SS7 messages. Set to DISABLED to put out of service.

Ss7AdrFlag — This flag is set if the message contains an SS7 address. Set to TRUE or FALSE.

Ss7AdrType — This field specifies the SS7 address type. Set to SS7_TCAP_ADR or SS7_TCC_ADR.

Ss7Adr — A data structure which may contain full SS7 addressing of DPC and OPC nodes. Full global title translation addressing is included in this format. See the header file **iu.h** for details.

If the Ss7AdrFlg field is not set, the field is ignored and SS7 addressing is taken from the default startup parameters of the SEPT process. When sending from a remote host, this address indicates where the signal is destined for. When received by an host, this addressing indicates where the signal originated from in the SS7 domain.

InetAdrFlag — This flag is set if the message contains an address. Set to TRUE or FALSE.

InetAdrType — This field specifies the type of address. Set to INET_ADR.

InetAdr — A data structure that specifies the address of an host in the domain. When operating SEPT in the TCP/IP mode, this field has little significance since the address will always reflect the current TCP connection. In the UDP mode, all SS7 subsystem messages sent by the remote host must always set this field.

MsgFlavor — A code indicating the protocol definition for “msg.”

MsgLen — Length of “msg” in bytes.

msg[1] — The beginning of the IUMSGX buffer containing information representing the various protocol flavors (i.e., TCAP, SCCP, etc.) being transported between SS7 and Ethernet.

7.2.3 Address Handling

Addressing is handled in such a way that allocation for host and SS7 addresses can be sent simultaneously in a single SS7 subsystem message. The host address flowing in a signal destined for the SEPT SEE/SS7 layer has no significance. However, flowing in the other direction with both addresses present, allows routing to an host and then identification by the host as to where on the SS7 network the signal originated. The host address represents the client host address the SS7 subsystem is attached to if the SS7 subsystem is acting as a TCP server.

```
#define     IU_TCAP_CCITT_TEMP 5
#define     IU_SCCP_ANSI_MNT    6
#define     IU_SCCP_ANSI_TEMP   7
#define     IU_SCCP_CCITT_MNT   8
#define     IU_SCCP_CCITT_TEMP  9

/* EBS DEPENDENT */
#define     L_TC_ABORT_INF_LENGTH       176
#define     REVISED_ABORT_LENGTH        16
#define     INITIAL_TC_OPR              0
#define     MSGMNB                      4096


/*** PORTICO CCITT TCAP PROTOCOL ADDRESSING FORMATS. ***/


        /* PORTICO CCITT SS7 TCAP, GLOBAL TITLE TYPE-1 ADDRESSING. */
typedef struct Ss7CcGt1Adr
{   BYTE    NatOfAdrInd;
    BYTE    AdrInfoLen;
    BYTE    AddrInfo[L_MAX_GT_ADDR_INFO];
} SS7CCGT1ADR;


        /* PORTICO CCITT SS7 TCAP, GLOBAL TITLE TYPE-2 ADDRESSING. */
typedef struct Ss7CcGt2Adr
{   BYTE    TransType;
    BYTE    AdrInfoLen;
    BYTE    AddrInfo[L_MAX_GT_ADDR_INFO];
} SS7CCGT2ADR;


        /* PORTICO CCITT SS7 TCAP, GLOBAL TITLE TYPE-3 ADDRESSING. */
typedef struct Ss7CcGt3Adr
{   BYTE    TransType;
    BYTE    EncodeScheme;
    BYTE    AdrInfoLen;
    BYTE    AddrInfo[L_MAX_GT_ADDR_INFO];
} SS7CCGT3ADR;
```

```
        /* PORTICO CCITT SS7 TCAP, GLOBAL TITLE TYPE-4 ADDRESSING. */
typedef struct Ss7CcGt4Adr
{   BYTE    TransType;
    BYTE    EncodeScheme;
    BYTE    NatOfAdrInd;
    BYTE    AdrInfoLen;
    BYTE    AddrInfo[L_MAX_GT_ADDR_INFO];
} SS7CCGT3ADR;


typedef struct TcCcAdr
{   BYTE DpaAdrInd;
    DWORD   dpc;
    BYTE    rssn;
    union
    {   SS7CCGT1ADR Gt1Adr;
        SS7CCGT2ADR Gt2Adr;
        SS7CCGT3ADR Gt3Adr;
        SS7CCGT4ADR Gt4Adr;
    } dpa;

    BYTE OpaAdrInd;
    DWORD   opc;
    BYTE    ssn;
    union
    {   SS7CCGT1ADR Gt1Adr;
        SS7CCGT2ADR Gt2Adr;
        SS7CCGT3ADR Gt3Adr;
        SS7CCGT4ADR Gt4Adr;
    } opa;
} SS7CCCADR;



        /* INET UDP/TCP ADDRESSING. */
typedef struct sockaddr_in  INETADR;


    /* THE UNIFIED PORTICO ADDRESSING MECHANISM. */
typedef union
{   SS7TCANADR      TcAnAdr;
    SS7TCCCADR      TcCcAdr;

    SS7ISANADR      IsAnAdr;
    SS7ISCCADR      IsCcAdr;
    X25CCADR        X25ScAdr;
    SDSADR          SdsAdr;
    INETADR         IpAdr;
} IUADR;

#define SS7_TCAN_ADR      1
#define SS7_TCCC_ADR      2
# define SS7_ISCC_ADR     4
#define X25_CC_ADR        5
#define SDS_ADR           6
#define INET_ADR          7
```

```
#define DEFLT_AN_ADR_IND  0x43
#define DEFLT_CC_ADR_IND  0x43


#define DPC_SSN_IND_MASK  3
#define DPC_SSN_PRESENT   3
#define SSN_PRESENT       2
#define DPC_PRESENT       1
```

7.2.4 ITU TCAP Protocol Flavor

The TcapCcFlvr section of the **iu.h** header defines a TCAP message format for the "msg" element of the IUMSGX message. This format allows ITU TCAP transactions between SS7 subsystem SEPT and host using the TCP/IP domain. The following data structures represent the ITU TCAP protocol flavor which is encapsulated in the SS7 subsystem signal.

```
/* THE PORTICO CCITT TCAP FLAVOR. */
typedef struct TcapCcFlv
{
        /* DIALOGUE SECTION */
    DWORD   DlgType;
    DWORD   DlgId;
    DWORD   Qserv;
    DWORD   CompPres;
    DWORD   termination;
    DWORD   AbortType;
    DWORD   AbortLen;
    BYTE    abort[L_TC_ABORT_INF_LENGTH];

        /* COMPONENT SECTION */
    DWORD   CompType;
    DWORD   InvokeIdFlg;
    DWORD   InvokeId;
    DWORD   LastComp;
    DWORD   TcClass;
    DWORD   LinkIdFlg;
    DWORD   LinkId;
    DWORD   TimeoutValue;

    DWORD   OprType;
    DWORD   LocalType;
    DWORD   GlobLength;
    BYTE    GlobIdent[L_GLBERR_INFO_LEN];

    DWORD   TypeErr;
    DWORD   LocalErr;
    DWORD   GlobErrLen;
    BYTE    GlobErr[L_GLBERR_INFO_LEN];

    DWORD   ProblemCode;
    DWORD   Problem;
    WORD    ParamType;
    WORD    ParamLength;
    BYTE    parameters[1];
}   TCAPCC;
```

The elements are as follows:

DlgType — This determines what kind of dialogue is going to be sent to the remote end. For ITU, the possibilities are:

> **L_TC_UNI** (0x61) — Request/indicate an unstructured dialog. Such a dialog occurs when a TC-user needs to send one or more components to a remote TC-user and replies are not expected.
>
> **L_TC_BEGIN** (0x62) — Use to establish a transaction with a remote peer transaction sub-layer. It may contain one or more components.
>
> **L_TC_CONTINUE** (0x65) — Continues a dialog.This is used following a Begin message to transfer additional information related to the transaction. The Continue message contains one or more components.
>
> **L_TC_END** (0x64) — This is used to terminate a transaction. All remaining untransmitted components are included in the End message.
>
> **L_TC_ABORT** (0xF6) — Allows a TC-user to terminate a dialog abruptly without transmitting any pending components.

DlgId — DlgId is a field that is essentially filled in by the SEPT TCAP API. For the TCP/IP user to obtain a dialogue resource ID from the SS7 subsystem TCAP API, the field must be initialized with 0.

If the user wants to perform multiple transactions on a single dialog ID, the value received from a SS7 subsystem signal response would be issued in outgoing transmission from the host.

Qserv — This field specifies which of the following quality of service types is expected by the TC-user from the SCCP layer:

> **L_SCCP_CNTLESS_NS_NR** (0x00) — Connectionless, Not Sequenced, No Return on error
>
> **L_SCCP_CNTLESS_S_NR** (0x01) — Connectionless, Sequenced, No Return on error
>
> L_SCCP_CNTLESS_NS_RE (0x80) — Connectionless, Not Sequenced, Return on Error
>
> **L_SCCP_CNTLESS_S_RE** (0x81) — Connectionless, Sequenced, Return on Error

Note that for ITU, this parameter is only to be filled for L_TC_UNI and L_TC_BEGIN. Once the dialog is started, the same quality of service parameter is used until the dialog ends.

CompPres — Not used.

termination — This field is used for building L_TC_END type dialog messages. It indicates which scenario is chosen by the TC-user for the end of the dialog. The possible values are:

> **L_BASIC_END** (0x00)
>
> **L_PRE_ARRANGED_END** (0x01)

AbortType — This field is used only with L_TC_ABORT type dialog messages. Unlike TCAP-generated abort messages, user-generated abort messages may contain any information element you find appropriate. To indicate that the abort cause is specifically given by the user, the AbortType has to be chosen as L_TC_U_ABORT__INF_TAG (0x6B). For the abort messages generated by the transaction sub-layer of the TCAP, this value is L_TC_P_ABORT_TAG (0x4A), which is returned if the incoming message is an abort message.

AbortLen — This field contains the length (in octets) of the user abort information. This value may not exceed L_ABORT_INF_LENGTH.

abort — This field is an array of octets holding the user abort information.

CompType — The component type may be one of the following:

L_TC_INVOKE (0xA1) — Invocation of an operation that may be linked to another operation.

L_TC_RESULT_L (0xA2) — The last part of a segmented result of a successfully executed operation.

L_TC_ERROR (0xA3) — Reply to a previously invoked operation indicating that the operation execution failed.

L_TC_REJECT (0xA4) — Rejection of a component by the TC-user indicating a malformation which prevents the operation from being executed or the replay from being understood.

L_TC_RESULT_NL (0xA7) — Non-final part of the segmented result of a successfully executed operation.

L_TC_U_CANCEL (0xAA) — This is used to terminate an operation invocation as a consequence of a TC-user decision.

InvokeIdFlg — Indicates whether an invoke identifier is present. Set to L_TC_FLAG_CLEAR.

InvokeId — Identifies an operation invocation. Set to 0 to 255.

LastComp — Flag indicating that the component being delivered is the last component. Set to L_COMPONENT_PRESENT or NO_MORE_COMPONENTS. Not used when sending.

TcClass — Component class can be one of the following:

L_CLASS_1 — (Both success and failure of the requested operation are reported)

L_CLASS_2 — (Only failure is reported)

L_CLASS_3 — (Only success is reported)

L_CLASS_4 — (Neither success nor failure is reported)

LinkIdFlg — This indicates whether a link identifier is present identifying a set of dialogs transpiring. Set to L_TC_FLAG_SET or L_TC_FLAG_CLEAR.

LinkId — Value identifying a set of dialogs transpiring.

TimeoutValue — This indicates the maximum lifetime of a component ID. It is used to handle cases in which operations do not receive an expected reply. It is optional; set to zero to disable.

SequenceTag — Not used.

OprType — The operation type is specified by one of the following:

L_TC_LOCAL_OP_CODE (0x02) — Local operation

L_TC_GLOBAL_OP_CODE (0x06) — Global operation

LocalType — Local type specifier. Set to 0 to 4,294,967,295.

GlobLength — Length of global operation identifier in bytes. The maximum length is L_GLBERR_INFO_LEN.

GlobIdent — Global operation identifier.

TypeErr — This can be L_TC_LOCAL_ERROR_CODE (0x02) or L_TC_GLOBAL_ERROR_CODE (0x06).

LocalErr — Local error type specifier.

GlobErrLen — Length of global error identifier.

GlobErr — Global error identifier.

ProblemCode — The categories of problem types are as follows:

L_TC_GEN_PROB (0x80) — General problem

L_TC_INVOKE_PROB (0x81) — Invoke problem

L_TC_RESULT_PROB (0x82) — Return result problem

L_TC_ERROR_PROB (0x83) — Return error problem

Problem — The specific problem within one of the categories above. Set to 0 to 4,294,967,295.

ParamType — TCAP parameter type. Set to 0 to 65535.

ParamLength — The length of the parameter information in bytes. The maximum length is L_TC_PAR_LENGTH.

parameters — An array containing the parameter information.

This mechanism is the key to allowing non-SS7 based equipment access to SS7 services without special configuration or additional kernel/hardware modifications. If a node on an TCP/IP network can communicate to SS7 subsystem over TCP/IP protocols, it has the ability to access and use SS7 services (i.e., TCAP/SCCP services with the SS7 subsystem).

7.2.5 ITU TCAP Template File

This template file contains a subset of the items found in the ITU TCAP flavor file.

```
    /* THE PORTICO CCITT TCAP TEMPLATE FILE. */
typedef struct TcapCcFile
{
        /* DIALOGUE SECTION */
    DWORD   DlgType;
    DWORD   DlgId;
    DWORD   Qserv;
    DWORD   CompPres;
    DWORD   termination;
    DWORD   AbortType;
    DWORD   AbortLen;
    BYTE    abort[L_TC_ABORT_INF_LENGTH];

        /* COMPONENT SECTION */
    DWORD   CompType;
    DWORD   InvokeIdFlg;
    DWORD   InvokeId;
    DWORD   LastComp;
    DWORD   TcClass;
    DWORD   LinkIdFlg;
    DWORD   LinkId;
    DWORD   TimeoutValue;
    DWORD SequenceTag;

    DWORD   OprType;
    DWORD   LocalType;
    DWORD   GlobLength;
    BYTE    GlobIdent[L_GLBERR_INFO_LEN];

    DWORD   TypeErr;
    DWORD   LocalErr;
    DWORD   GlobErrLen;
    BYTE    GlobErr[L_GLBERR_INFO_LEN];

    DWORD   ProblemCode;
    DWORD   Problem;
    WORD    ParamType;
    WORD    ParamLength;

    BYTE    parameters[L_TC_MAX_PAR_LENGTH];
}   TCAPCCFL;
```

7.2.6 TCAP Template Protocol Flavor

The data structures in this section represent the TCAP template protocol flavor. Templates further remove SS7 details that are unnecessary at the host application level. Templates are essentially a subset of contents found in its related protocol flavor counterpart. In the case of the ITU TCAP template, its superset relation would be the ITU TCAP protocol flavored structures.

Templates contain only the absolute necessities of a full TCAP structure that an application needs to carry out basic services. The other fields not directly accessible through the data structures can be statically defined in a text file or files on the SS7 subsystem.

The SS7 subsystem will handle the process of merging the template information and statically defined related information residing in SEPT into a full TCAP message ready for transport. Key points to note with templates are that all fields are of fixed length except the parameter field. The same rules apply for allocation of space as is for the full service formats.

```
  /* THE PORTICO CCITT TCAP TEMPLATE FLAVOR. */
typedef struct IuTcapCcTemp
{   DWORD   TempId;
    DWORD   LinkEdIdFlg;
    DWORD   LinkEdId;
    DWORD   InvokeIdFlg;
    BYTE    InvokeId;
    DWORD   DlgId;
    DWORD   ParamFlag;
    DWORD   ParamLength;
    BYTE    parameters[1];
} TCAPCCTEMP;
```

7.2.7 ITU SCCP Maintenance Flavor

The SCCP protocol flavor is used to transfer SCCP Maintenance information to the remote host. Operation status involving remote DPC:SSN application service states as well as network failures are made available through this mechanism. Specifically, signaling in this format is generated for the following events:

- Destination Point Code Out of Service
- DPC:SSN Application Out Of Service
- DPC Resumed Service
- DPC:SSN Application has Resumed Service
- Network Failure
- Network Congestion/Network Failure on Transmission

```
    /* THE PORTICO ANSI/CCITT SCCP MAINTENANCE FLAVOR. */
typedef struct IuSccpMntFlv
{   DWORD       MsgType;
    DWORD       status;
    DWORD       multiplicity;
    DWORD       ReturnReason;
    IPCmsg_t    SccpMsg;
} SCCPMNT;
```

MsgType — This specifies the type of SCCP maintenance message being transported:

N_STATE — Indicates a state change of a subsystem of a known point code.

N_PCSTATE — Indicates a state change in status of a known point code.

N_NOTICE — Indicates that a message sent from TCAP to SCCP cannot be transported.

dpc — Destination Point Code in integer format.

ssn — Subsystem number of known point code application.

status — True indicates that the SCCP is still in service. A False indication means that the SCCP is not in service.

multiplicity — Identifier specifying TCAP application if multiple invocations are active using the same DPC:SSN addressing information.

ReturnReason — Specifies a failure corresponding to an "N_NOTICE" message:

L_UDTRET_no_xlate_for_spec_addr — Bad DPC/SSN address

L_UDTRET_no_xlate_for_addr — Bad DPC/SSN address

L_UDTRET_subsys_congestion — Message congestion for the specified subsystem address

L_UDTRET_subsystem_failure — Failure of subsystem

L_UDTRET_unequipped_user — Unequipped or bad user part

L_UDTRET_network_failure — Network failure

L_UDTRET_network_congestion — Network congested

L_UDTRET_unqualified — Unqualified

IPCmsg_t — Raw SCCP message. Generally all information needed by the application layer can be obtained from the previously defined fields. The complete SCCP message is provided here. However, more information is required for remote access.

7.3 BUILDING THE TCAP COMPONENT

Each SS7 subsystem TCAP message contains a TCAP dialogue section and possibly a TCAP component section. These groups of information must be constructed based on tables that follow. The data structures of the SS7 subsystem TCAP message flavors found in the header file **iu.h**.

To build a TCAP component, select the component type of interest and then determine the fields specified as mandatory or optional. The fields will need to be filled out in the SS7 subsystem TCAP message structure in the case of sending a TCAP message. If receiving a TCAP message, the specified fields would be read.

Table 7.1: ITU Component Structures

Component	TC-INVOKE	TC-RESULT-L TC_RESULT-NL	TC-UERROR	TC-UREJECT	TC-UCANCEL
CompType	M	M	M	M	M
DlgId	M	M	M	M	M
InvokeId	M	M	M	M	M
LastComp	(-)	(-)	(-)	(-)	NA
ParamLength	M	M	M	M	NA
Parameters	O	O	O	O	NA
UNION TYPE	invoke	result	error	reject	–
class	M	NA	NA	NA	NA
LinkIdFlg	M	NA	NA	NA	NA
LinkId	O	NA	NA	NA	NA
Time-outValue	M	NA	NA	NA	NA
OprType	M	O	NA	NA	NA
sequence_tag	NA	M	NA	NA	NA
ErrorType	NA	NA	M	NA	NA
local/global	NA	NA	M	NA	NA
Problem-Code	NA	NA	NA	M	NA
ProblemSpec	NA	NA	NA	M	NA

M:Mandatory/O:Optional/NA:Not Applicable/(-) — Parameters returned to the user by TC_getcmp

7.3.1 Component Field Descriptions

This section contains brief descriptions of the various component fields. For a complete description, refer to the ITU recommendations Q771-Q775.

L_TC_INVOKE (0xA1) — Invocation of an operation to be performed. This may be linked to another operation sent from the other end.

L_TC_RESULT-L (0xA2) — The last part of a segmented or only result of a successfully executed operation.

L_TC_RESULT_NL (0xA7) — Non-final part of the segmented result of a successfully executed operation.

L_TC_ERROR (0xA3) — Reply to a previously invoked operation, indicating that the operation execution failed.

L_TC_REJECT (0xA4) — Rejection of a component by the TC-user, indicating a malformation which prevents the operation from being executed or the reply from being understood.

L_TC_U_CANCEL (0xAA) — Used to terminate an operation invocation as a consequence of a TC-user decision.

DlgId — The dialog ID which relates the component built TC_putcmp to a specific dialog.

InvokeId — This is an integer value which identifies an operation invocation.

LastComp — This is not used by TC_putcmp. It is set if the component given to the user by the TC_getcmp is the last component in a TCAP message.

ParamLength — The length of the parameter region in octets.

parameter — An array of octets holding the user data. There is no restriction on the structure of the parameter block except that the length of the region cannot exceed L_TC_MAX_PAR_LENGTH.

TcClass — This is used in a L_TC_INVOKE type component and is one of the following:

- **L_CLASS_1** (both success and failure of the requested operation are reported by the remote user)
- **L_CLASS_2** (only failure is reported)
- **L_CLASS_3** (only success is reported)
- **L_CLASS_4** (neither success nor failure is reported)

LinkIdFlg — This indicates whether a linked ID is present in the “linkid” field of the “CMP_s” component structure. If LinkIdFlg is not set, the linkid field is not considered by TC_putcmp.

LinkId — This links an operation invocation to a previous operation invocation which is still active. The invoke ID of the received component is reflected in “linkid,” in case a linked operation to an incoming operation is to be generated.

TimeoutValue — This indicates the maximum lifetime of a component ID (i.e. invoke ID). It is used to handle cases in which operations do not receive any expected reply. This is mandatory in ITU. If entered as 0, the feature is disabled.

OprType — This stands for operation and identifies the action to be executed by the remote TC-user. It is of type “opr_t” which is defined in the tcap.h header file. Two types of operation are possible: Local operation and Global operation. The “type” field of the “OprType” must be filled with L_TC_LOCAL_OP_CODE (0x02) or L_TC_GLOBAL_OP_CODE (0x06) respectively. The “local” and “global” fields must also be filled respectively. The “global” is of type “global_t” which is a small data structure that is suitable for holding the global operation codes described in ITU recommendation X.209.

sequence_tag — This is used as a flag to indicate that there is a sequence or set of parameters (L_TC_PARAMETER_SEQ_TAG/0x30 or L_TC_PARAMETER_SET_TAG/0x31) to be assigned to sequence_tag accordingly.

ErrorType — This is used for TC_U_ERROR type components only and specifies whether the error is of local (L_TC_LOCAL_ERROR_CODE/0x02) or global (L_TC_GLOBAL_ERROR_CODE/0x06) type.

local/global — Depending on the type of error, either the “local” or the “global” field must be filled. “local” is of DWORD type and “global” is of type “global_t” which is defined in the “tcap.h” header file.

ProblemCode — Used only for L_TC_U_REJECT type components, this holds the information about why the received component is being rejected. It is one of the following:

- **L_TC_GEN_PROB** (general problem)
- **L_TC_INVOKE_PROB** (invoke problem)
- **L_TC_RESULT_PROB** (return result problem)
- **L_TC_ERROR_PROB** (return error problem)

ProblemSpec — The possible problem types for each of the above problem categories are defined in the "tcap.h" header file. TC_putcmp returns values 0 and -1 for success and failure respectively and "error" is set to the following values:

- **EDLGID** (invalid dialogue ID)
- **EMSGSIZE** (reserved buffer size is not large enough)
- **ECMPTYPE** (invalid component type)
- **ECMPPAR** (error in component parameters)
- **ENOBU** (no more free buffers)
- **EINVLID** (invalid link or correlation ID [i.e., the operation to which another operation tried to link does not exist])

7.4 BUILDING THE TCAP DIALOGUE

To build a SS7 subsystem TCAP dialog, select the dialog type of interest and then determine the fields specified as mandatory or optional. These fields need to be filled out in the SS7 subsystem TCAP message structure in the case of sending a TCAP message. If receiving a TCAP message, the specified fields would be read.

Table 7.2: ITU Dialogue Structure

Dialogue_t	TC_UNI	TC_BEGIN	TC_CONTINUE	TC_END	TC_U_ABORT
DlgType	M	M	M	M	M
DlgId	M	M	M	M	M
UNION TYPE	uni	begin	cont	end	abort
Qserv	M	M	M	M	M
DPA	M	M	NA	NA	NA
OPA	M	M	NA	NA	NA
CompPres	(-)	(-)	(-)	(-)	NA
termination	NA	NA	NA	M	NA
AbortType	NA	NA	NA	NA	O
AbortLen	NA	NA	NA	NA	O
abort	NA	NA	NA	NA	O

M:Mandatory/O:Optional/NA: Not Applicable/(-) — Parameters returned to the user by TC_getdlg

7.4.1 Field Descriptions

DlgType — This field determines what kind of dialog is to be sent to the remote end. The following are the values for DlgType:

L_TC_UNI (0x61) — Requests/indicates an unstructured dialog. Such a dialog occurs when a TC-user needs to send one or more components to a remote TC-user and replies are not expected.

L_TC_BEGIN (0x62) — Use to establish a transaction with a remote peer transaction sub-layer. It may contain one or more components.

L_TC_CONTINUE (0x65) — Continues a dialog. This is used following a Begin message to transfer additional information related to the transaction. The Continue message contains one or more components.

L_TC_END (0x64) — This is used to terminate a transaction. All remaining untransmitted components are included in the End message.

L_TC_ABORT (0x67) — Allows a TC-user to terminate a dialog abruptly without transmitting any pending components.

DlgId — Determines for which dialog ID the message is being built. The components with this dialog ID will be put into this TCAP message.

Qserv — This field specifies which of the following quality of service types is expected by the TC-user from the SCCP layer:

L_SCCP_CNTLESS_NS_NR (0x00) — Connectionless, Not Sequenced, No Return on error

L_SCCP_CNTLESS_S_NR (0x01) — Connectionless, Sequenced, No Return on error

L_SCCP_CNTLESS_NS_RE (0x80) — Connectionless, Not Sequenced, REturn on error

L_SCCP_CNTLESS_S_RE (0x81) — Connectionless, Sequenced, REturn on error

Note that this parameter is only to be filled for L_TC_UNI and L_TC_BEGIN (i.e., when a dialog is being started). Once the dialog is started, the same quality of service parameter is used until the dialog ends.

DPA/OPA — These are the called and calling party addresses:

DPA (Destination Point Address/called party address)

OPA (Originating Point Address/calling party address)

CompPres — This is set by the TC_rcvdlg function if there are no components present in the incoming TCAP message. It is a (-) type field, which means you do not have to fill it in. It is returned in the “dlg_s” structure when TC_rcvdlg call is made. TC_snddlg calls ignore this field.

termination —This field is used for building L_TC_END type dialog messages. It indicates which scenario is chosen by the TC-user for the end of the dialog. Two possible values exist:

L_BASIC_END (0x00)

L_PRE_ARRANGED_END (0x01)

AbortType — This field is used only with L_TC_ABORT type dialog messages. Unlike TCAP-generated abort messages, user-generated abort messages may contain any information element you find appropriate. To indicate that the abort cause is specifically given by the user, the abort_type has to be chosen as L_TC_U_ABORT_INF_TAG (0xD8). (For the abort messages generated by the transaction sub-layer of the TCAP, this value is L_TC_P_ABORT_TAG (0xD7), which is returned by the TC_rcvdlg call if the incoming message is an abort message.)

AbortLen — This field contains the length (in octets) of the user abort information. This value may not exceed L_ABORT_INF_LENGTH.

abort — This field is an array of octets holding the user abort information. TC_snddlg returns the following values:

0 (on success) No more components to be sent

1 (on success) More components exist to send

-1 (on failure) “error” is set to the following possible errors:

- **EDLGID** (Invalid dialog ID is given to TC_snddlg)
- **EDLGTYP** (Invalid dialog type)
- **EDLGPAR** (Error in dialog parameter)

7.5 TCP/IP ADDRESSING

When invoking SS7 subsystem SEPT within TCP/IP client or server mode, a virtual connection is made between SEPT and the TCP/IP host process. As a result, the SEPT SCL layer has no control of the TCP/IP addressing since the address cannot be changed or modified when in service.

Note that before releasing, a host is responsible for placing the SEPT process in the disabled state. This will generate an SCCP message to the far end indicating that the subsystem application is out of service. The newly connected host must then put SEPT in the Active state. This will also generate an SCCP message indicating that the application is in service.

Figure 7.2 shows the addressing scenario for a TCP connection-oriented transfer. The server is started first and waits until a client establishes a connection by connecting to the server.

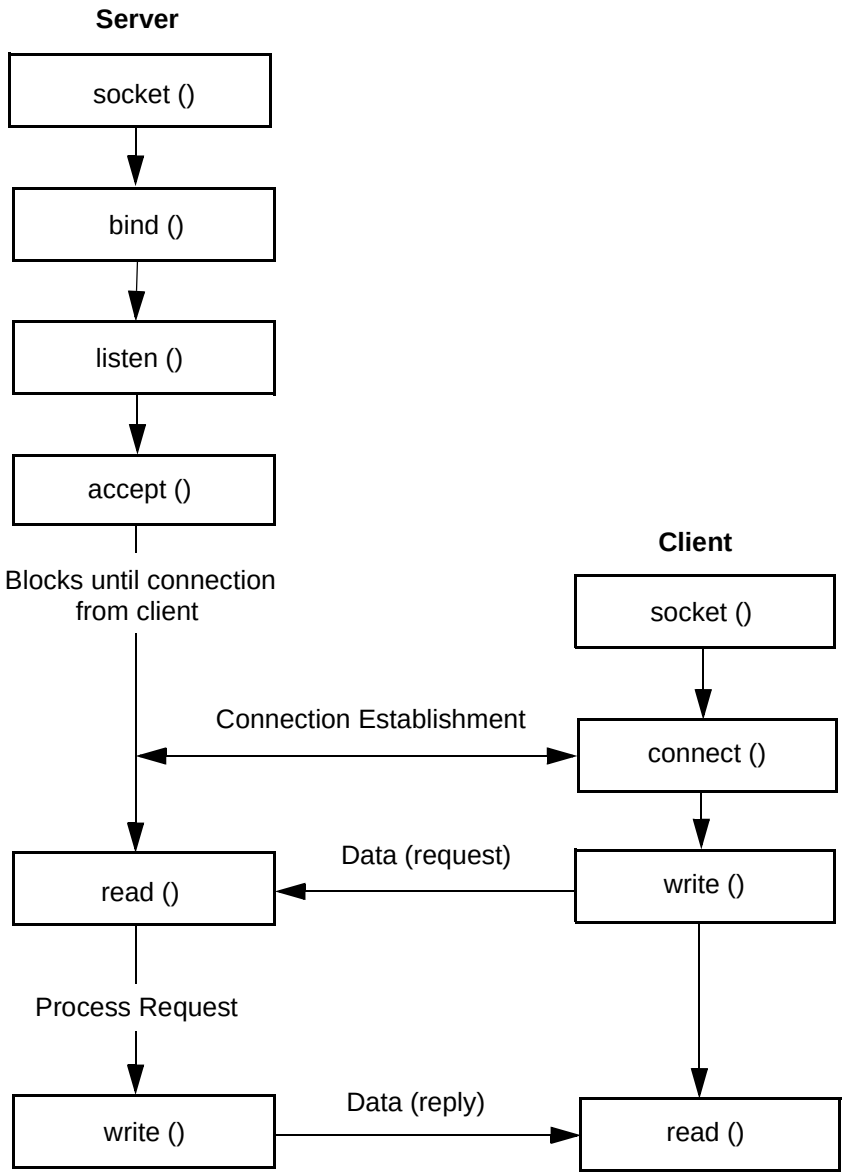


Figure 7.2: Socket System Calls For TCP Protocol

7.6 SS7 SUBSYSTEM TEMPLATES

The SS7 subsystem signaling packets discussed in *Section 7.2* remove the need for the application to directly manipulate protocol stacks at the local system call level. However, there still may be detail and overhead encoded within the flavored message structure that the application programmer may wish to off-load and deal with elsewhere.

Templates are mechanisms supported by the SS7 subsystem SEPT which further reduce the detail found in a flavored protocol message packet. Templates have all details of a protocol message stored in a file structure. The template, being a subset of its more sophisticated relative, is then used which contains only the bare minimum of dialog/component information needed for applications to provide meaningful services to and from the target protocol stack.

To use a template, simply specify a template ID number when commanding an operation to a remote protocol stack. The ID number is interpreted and the correct template file is used to fill in the details about how the resulting TCAP/SS7message is to be constructed.

Another benefit of the templates is that parameters for SS7 transport can vary considerably without having to make any source code changes at the application level. Refer to *Section 7.8*for examples of the SEPT template files.

7.6.1 Using The Templates

SS7 subsystem TCAP templates consist of essentially two components. The first is an ASCII file that contains the bulk of the TCAP elements used to build a TCAP/SS7 message. The format of this file is a C data structure which maps directly into the SS7 subsystem TCAP format specified in the header file **iu.h**.

This “template file” is compiled down into a runtime format through the use of utilities supplied with the SS7 subsystem product. At runtime, this image of the TCAP template file is loaded into core by SEPT, where it stands ready to be accessed as an element in a database of other TCAP templates for TCAP message synthesis.

The second component of the SS7 subsystem template is the “TCAP Template Signal.” This is a unique protocol flavor carried in the SS7 subsystem signal that contains an identifier or index that specifies the TCAP template databases as well as a subset of information elements found in an SS7 subsystem TCAP structure.

This subset of TCAP elements are those that are almost always needed at the application level to carry out TCAP/SS7 services for the application.

When using the TCAP template feature, signaling is generated at the host using the TCAP template signal format. The signal specifies the remainder of the TCAP information through it's "template file identifier." The template signal and corresponding TCAP information are then synthesized and assembled with SEPT. This results in a TCAP/SS7 message. The significance of this is that the application level did not have to involve itself with details of TCAP elements that do not have any direct bearing on the application. Another added benefit is that the TCAP message structure and content can be changed through the TCAP template file; this eliminates application source code reworking and recompiling.

The following is an outline of the process needed to generate a working TCAP Template Database File:

1. Locate the default TCAP template database file, **TcCcFls**, located in the **$SNV** directory. This file contains the names of six example executable TCAP template files.
2. Using a UNIX editor such as vi, you may modify the default template database file found in Step 1 (**TcCcFls**) by renaming the sample files or adding more template files. (Refer to the template file example in this section.)
3. If you chose to rename or add any template files in the default template database file, you must also modify the filenames found in **$TC/makefile.TcTemps** (**$TC** is defined as **/home/SEPT/sept-tx/TcTemps**) to reflect your chosen TCAP template filenames. Use a UNIX editor such as vi.
4. Compile your file from Step 3 to generate executable TCAP template files by typing the following and pressing **Return** after each line.

   cd $TC

   make -f makefile.TcTemps

   *NOTE: You may compile on any platform, but the results of the **make -f** command must be able to run in a Solaris environment.*

   *To use your own compiler, you must first change the compiler path in the **makefile.TcTemps** file.*
5. Copy the resulting file(s) from Step 4 to directory **$SNV** by typing the following and pressing **Return** for each file:

   copy <filename> $SNV/

   where **filename** is an executable TCAP template file generated in Step 4.

The following table defines the sample TCAP template files found in **$SNV/TcCcFls**.

Table 7.3: SS7 Subsystem Template Files

Template File	Description
TcCcBeg	TCAP ITU begin
TcCcEnd	TCAP ITU end
TcCcUniInv	TCAP ITU unidirectional invoke
TcCcUniResp	TCAP ITU unidirectional response
TcCcCon	TCAP ITU continue
TcCcAbort	TCAP ITU abort

7.7 EXAMPLE

A print out of the ITU template files TcBeg.c and TcCcEnd.c follow:

TcCcBeg.c

```
static char rcsid[]="$Id: TcCcBeg.c,v 1.1 1994/07/07 18:04:48 manoj Exp
manoj-tc $";


/* TcCcBeg.c- CCITT TCAP TEMPLATE SPECIFIYING A "BEGINS A DIALOG" TCAP
MESSAGE.*/
/* It should be noted that CCITT TCAP dialog types are different from ANSI TCAP
*/
/* The names of the template files have been in accordance to the
dialog_type (dlg_type) as specified in CCITT blue books */



#define  TCAP_CCITT
#define  CCITT
#include<TcTemplates.h>



char TcTempFlNm[] = {"TcCcBeg.temp"};

TCAPCCFLTcTemplate =
{
                              /* DIALOGUE SECTION. */
/* DIALOGUE TYPE */L_TC_BEGIN,
/* DIALOGUE ID */ 0,


/* QUALITY OF SERVICE */L_SCCP_CNTLESS_NS_RE,
```

```
/* COMPONENT PRESENT */    NA,
/* TERMINATION*/    NA,
/* ABORT TYPE */    NA,
/* ABORT LENGTH */0,
/* ABORT */          {
                           0
                           },

                           /* COMPONENT SECTION. */
/* COMPONANT TYPE*/ L_TC_INVOKE,
/* INVOKE ID FLAG */TRUE,
/* INVOKE ID */    VAR,
/* LAST COMPONANT */NA,
/* CLASS */          L_CLASS_1,

/* LINK ID FLAG */FALSE,
/* LINK ID */      NA,
/* TIMEOUT VALUE */0,
/* SEQEUNCE TAG */NA,

/* OPERATION TYPE */L_TC_LOCAL_OP_CODE,
/* LOCAL TYPE OPR. */NA,
/* GLOBAL OPR LENGTH */NA,
/* GLOBAL IDENTIFIER */{
                           NA
                           },

/* TYPE ERROR */NA,
/* LOCAL ERROR*/NA,
/* GLOBAL ERROR LENGTH */NA,
/* GLOBAL ERROR */{ NA },

/* PROBLEM CODE */NA,
/* PROBLEM*/         NA,

/* PARAMTER TYPE */ L_TC_PARAMETER_SET_TAG,
/* PARAMTER LENGTH */ 10,
/* PARAMTER STRING */{
                           1,2,3,4,5,6,7,8,9,0
                           }
};
#include"TcTempOpr.c"
```

TcCcEnd.c

```
static char rcsid[]="$Id: TcCcEnd.c,v 1.1 1994/07/07 18:04:48 manoj Exp
manoj-tc $";

/* TcCcEnd.c- CCITT TCAP TEMPLATE SPECIFIYING AN "ENDS A DIALOG" TCAP
MESSAGE.*/
/* It should be noted that CCITT TCAP dialog types are different from ANSI TCAP
*/
/* The names of the template files have been in accordance to the
dialog_type (dlg_type) as specified in CCITT blue books */



#define  TCAP_CCITT
#define  CCITT
#include<TcTemplates.h>



char TcTempFlNm[] = {"TcCcEnd.temp"};

TCAPCCFLTcTemplate =
{
                              /* DIALOGUE SECTION. */
/* DIALOGUE TYPE */L_TC_END,
/* DIALOGUE ID */ 0,

/* QUALITY OF SERVICE */L_SCCP_CNTLESS_NS_RE,
/* COMPONENT PRESENT */    NA,
/* TERMINATION*/    L_BASIC_END,
/* ABORT TYPE */    NA,
/* ABORT LENGTH */0,
/* ABORT */           {
                              0
                              },

                              /* COMPONANT SECTION. */
/* COMPONANT TYPE*/ L_TC_RESULT_L,
/* INVOKE ID FLAG */TRUE,
/* INVOKE ID */    VAR,
/* LAST COMPONANT */NA,
/* CLASS */           L_CLASS_1,

/* LINK ID FLAG */FALSE,
```

```
/* LINK ID */      NA,
/* TIMEOUT VALUE */0,
/* SEQEUNCE TAG */NA,

/* OPERATION TYPE */L_TC_LOCAL_OP_CODE,
/* LOCAL TYPE OPR. */NA,
/* GLOBAL OPR LENGTH */NA,
/* GLOBAL IDENTIFIER */{
                            NA
                            },

/* TYPE ERROR */NA,
/* LOCAL ERROR*/NA,
/* GLOBAL ERROR LENGTH */NA,
/* GLOBAL ERROR */{ NA },

/* PROBLEM CODE */NA,
/* PROBLEM*/          NA,

/* PARAMTER TYPE */ L_TC_PARAMETER_SET_TAG,
/* PARAMTER LENGTH */ 10,
/* PARAMTER STRING */{
                            "abcdefghij"
                            }
};


#include"TcTempOpr.c"
```

7.8 REMOTE HOST DEMONSTRATIONS

The following demonstration programs show basic techniques that would have to be implemented by host programs when interworking with SEPT. These basic demonstrations could be taken as is and used to provide basic skeletons for real host programs. All major operations of SEPT are demonstrated and include:

- Establishment of the TCP/IP connection
- Setup of the template signal
- Transmission and reception of SS7 subsystem signals
- Processing of SS7 subsystem signal information

7.8.1 Host Initiating a Query

The sample program below shows how to interface to SEPT as a TCP/IP client. The program generates queries through SS7 subsystem “template signals” and receivers responses from SEPT.

```
static char rcsid[]="$ID";

/*****************************************************************************
*                                                                            *
* Copyright (c) 1992 Summa Four Inc.                                         *
* All rights reserved.                                                       *
*                                                                            *
* This document contains confidential and proprietary information of         *
* Summa Four and any reproduction, disclosure, or use in whole or in part    *
* is expressly prohibited, except as may be specifically authorized by prior *
* written agreement or permission from Summa Four.                           *
*                                                                            *
*****************************************************************************
* VERSION      :                                                             *
*                                                                            *
* MODULE NAME  : invcl.c                                                     *
* DESCRIPTION  : THIS FILE CONTAINS SOURCE TO AN EXAMPLE WHICH SHOWS HOW TO  *
*                INTERFACE TO THE PORTICO SEPT AS A TCP/IP CLIENT. THE       *
*                EXAMPLE SENDS AND RECEIVES QUERIES AND RESPONSES            *
*                SIMULTANIOUSLY, PRINTING THE RESPONSES AS THEY ARE          *
*                RECEIVED. OUTGOING QUERY FORMAT IS ESTABLISHED THROUGH THE  *
*                USE OF A TEMPLATE FILE.                                     *
*                                                                            *
*                THIS IS A CLASSIC EXAMPLE OF HOW APPLICATIONS WORKING AS    *
*                PORTICO REMOTE HOST WOULD INTERFACE TO THE SUMMA FOUR      *
*                SEPT PRODUCT LINE.                                          *
*                                                                            *
*****************************************************************************
*                                                                            *
*                      RESTRICTED RIGHTS LEGEND                              *
* Use, duplication, or disclosure by Government Is Subject to restrictions as *
* set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and   *
* Computer Software clause at DFARS 252.227-7013                             *
*                                                                            *
*****************************************************************************/
```

```
#define     ANSI
#define     ERROR - -1
#define     ALWAYS  1
#define     TRUE    1
#define     FALSE   0

#include    <stdio.h>
#include    <sys/types.h>
#include    <sys/limits.h>
#include    <sys/ipc.h>
#include    <sys/msg.h>
#include    <sys/sem.h>
#include    <sys/signal.h>

#include    <sys/socket.h>
#include    <netinet/in.h>
#include    <arpa/inet.h>
#include    <netdb.h>

#include    <api.h>
#include    <scpa.h>
#include    <sccl.h>
#include    <scprims.h>
#include    <sccp_macros.h>
#include    <tcap.h>
#include    <iu.h>

#define     TCTEMPLATE1     0
#define     STRING          "0123456789"
typedef struct sockaddr     SOCKADDR;
typedef struct sockaddr_in  SOCKADDR_IN;
typedef struct hostent      HOSTENT;


extern int  errno;
HOSTENT     *HostByNm;
u_short     IuPort;
int         SockFd, QuerysPerSec, MsgSz;
IUMSGX      *XmtMsgX, *RcvMsgX;
TCAPANTEMP  *XmtTcTempMsg, *RcvTcTempMsg;
char        XmtMsgBfr[sizeof (IUMSGX) + sizeof (TCAPAN) + sizeof (STRING)];
char        RcvMsgBfr[sizeof (IUMSGX) + sizeof (TCAPAN) + sizeof (STRING)];


main (argc, argv)
int     argc;
char    **argv;
{
SOCKADDR_IN    IuAdr;
void           cleanup(), (*signal())(), GenQuery();
int            rcount, InvokeId, readn();
HOSTENT        *gethostbyname();

                /* SEE THAT INCOMING ARGUMENT COUNT IS CORRECT. */

    if (argc < 4)
    {   printf ("USAGE: invcl [ServerName] [ServerPort] [QuerysPerSecond]\n");
        exit (0);
    }
```

```
            /* SETUP TO RECEIVE TERMINATION SIGNALS. */

if (CatchSigs() == ERROR)
{   printf ("invcl: main()- Failed trying to setup SIGNALS...");
    exit (1);
}
            /* GET THE HOST BY IT'S NAME AND SETUP THE SOCKET
               STRUCTURE FOR THE SERVER PROCESS. */

if ((HostByNm = gethostbyname (argv[1])) == 0)
{   fprintf (stderr, "invcl: main- FAILED getting host name...\n");
    exit (1);
}
IuPort = atoi (argv[2]);
QuerysPerSec = atoi (argv[3]);

memset((char *)&IuAdr, 0, sizeof(SOCKADDR_IN));
bcopy(HostByNm->h_addr, (char*)&IuAdr.sin_addr, HostByNm->h_length);
IuAdr.sin_family = AF_INET;
IuAdr.sin_port = htons (IuPort);

            /* CREATE A TCP/IP SOCKET. */

if ((SockFd = socket (AF_INET, SOCK_STREAM, 0)) < 0)
{   fprintf (stderr, "invcl: main()- Couldn't OPEN socket...");
    exit (0);
}
printf ("invcl: main()- Socket created... \n");

            /* CONNECT TO THE TCP SERVER. */

if (connect (SockFd, (SOCKADDR *)&IuAdr, sizeof (IuAdr)) < 0)
{   fprintf (stderr, "invcl: main- FAILED on CONNECT...\n");
    exit (0);
}
printf ("invcl: main()- Connected...\n");

            /* SETUP POINTER FOR THE TRANSMIT AND RECEIVED BUFFERS
               WHICH HOLD OUTGOING AND INCOMING SEPT MESSAGES. */

XmtMsgX = (IUMSGX*)XmtMsgBfr;
XmtTcTempMsg = (TCAPANTEMP*)XmtMsgX->msg;
RcvMsgX = (IUMSGX*)RcvMsgBfr;
RcvTcTempMsg = (TCAPANTEMP*)RcvMsgX->msg;

            /* SETUP SIGNAL TO KICK THE QUERY GENERATOR INTO ACTION. */

if (signal (SIGALRM, GenQuery) == BADSIG)
{   fprintf (stderr, "invcl: main()- ERROR on SIGNAL initialization...\n");
    exit (1);
}

            /* INTIALIZE THE OUTGOING SEPT MESSAGE HEADER. */

XmtMsgX->HdrId = htons (GENERIC_SS7IU_HDR);
XmtMsgX->SessionId = 0;
XmtMsgX->state = htons (TRUE);
XmtMsgX->Ss7AdrFlag = htons (FALSE);
XmtMsgX->MsgFlavor = htons (IU_TCAP_ANSI_TEMP);
XmtMsgX->MsgLen = htons (sizeof (TCAPANTEMP) + sizeof (STRING) - 1);
```

```
                /* INITIALIZE THE SEPT ANSI TCAP TEMPLATE USED FOR OUTGOING
                   SEPT SIGNALING. */

    XmtTcTempMsg->TempId = htons (TCTEMPLATE1);
    XmtTcTempMsg->DlgId = 0;
    XmtTcTempMsg->LinkEdIdFlg = XmtTcTempMsg->LinkEdId = 0;
    XmtTcTempMsg->InvokeIdFlg = htons (TRUE);
    XmtTcTempMsg->InvokeId = InvokeId = 0;
    XmtTcTempMsg->ParamFlag = htons (TRUE);
    XmtTcTempMsg->ParamLength = htons (sizeof (STRING) - 1);
    strncpy (XmtTcTempMsg->parameters, STRING, sizeof (STRING));
   MsgSz = sizeof (IUMSGX) + sizeof (TCAPANTEMP) + ntohs (XmtTcTempMsg->ParamLength);

                /* PRIME THE PUMP OF THE QUERY GENERATOR THEN BLOCK
                   WAITING FOR RESPONSES RESULTING FROM QUERIES
                   MADE BY THE GENERATOR. */
    alarm (1);
    for (; ALWAYS; XmtTcTempMsg->InvokeId = htons (++InvokeId))
    {
                /* ALWAYS READ THE FIXED LENGTH SEPT MESSAGE HEADER FIRST.
                   IT WILL TELL YOU THE SEPT SS7 PROTOCOL FLAVOR THAT
                   FOLLOWS AS WELL AS IT'S LENGTH. */

    if ((rcount = readn (SockFd, RcvMsgBfr, sizeof(IUMSGX))) == ERROR)
        {   fprintf (stderr, "invcl: main()- FAILURE on read of socket...");
            continue;
        }
        else if (rcount == 0)
        {   printf ("invcl: main()- TCP connection has been terminated by server.
   Exiting...\n");
            cleanup();
            exit (1);
        }
                /* READ THE VARIABLE PROTOCOL FLAVOR PORTION OF THE SEPT
                   MESSAGE AND PRINT IT. */

       if ((rcount = readn (SockFd, RcvMsgBfr + sizeof (IUMSGX), ntohs (RcvMsgX->Ms-
   gLen))) == ERROR)
        {   fprintf (stderr, "invcl: main()- FAILURE on read of socket...\n");
            continue;
        }
        else if (rcount == 0)
        {   printf ("invcl: main()2- TCP connection has been terminated by server.
   Exiting...\n");
            cleanup();
            exit (1);
        }
        /*printf ("invcl: main()- Response received...\n");*/
        /*IuMsgPrint (RcvMsgX);*/
    }
}
```

```
/* GenQuery-

   GENERATES THE OUTGOING QUERIES TO SEPT. FUNCTION GETS CALLED BY AN ALARM SIGNAL.
   THE RATE IS DETERMINED BY A PREVIOUSLY DEFINED STATIC. FORMATS FOR THE
   OUTGOING MESSAGE MUST ALREADY BE ESTABLISHED. */

void GenQuery ()
{
    int     i, writen();

    for (i = 0; i < QuerysPerSec; i++)
    {   if (writen (SockFd, XmtMsgBfr, MsgSz) == ERROR)
            fprintf (stderr, "invcl: GenQuery()- FAILURE on write of socket...");
        htons (ntohs (XmtTcTempMsg->InvokeId)++);
    }
    /*printf ("invcl: GenQuery()- %d QUERYS written...\n\n", QuerysPerSec);*/

    if (signal (SIGALRM, GenQuery) == BADSIG)
    {   fprintf (stderr, "invcl: GenQuerys()- ERROR on SIGNAL initialization...\n");
        exit (1);
    }
    alarm (1);
}
```

7.8.2 Host Responding to a Query

The following program demonstrates an host application responding to SEPT query signals with SEPT response signals.

```
static char rcsid[]="$ID";

/*****************************************************************************
*                                                                            *
* Copyright (c) 1992 Summa Four Inc.                                         *
* All rights reserved.                                                       *
*                                                                            *
* This document contains confidential and proprietary information of         *
* Summa Four and any reproduction, disclosure, or use in whole or in part    *
* is expressly prohibited, except as may be specifically authorized by prior *
* written agreement or permission from Summa Four.                           *
*                                                                            *
*****************************************************************************
* VERSION      :       *
*                                                                            *
* MODULE NAME  : respcl.c                                                    *
* DESCRIPTION  : THIS FILE CONTAINS SOURCE TO AN EXAMPLE WHICH SHOWS HOW TO  *
*                INTERFACE TO THE PORTICO SEPT AS A TCP/IP CLIENT. THE       *
*                EXAMPLE RECEIVES AND SENDS QUERIES AND RESPONSES            *
*                SIMULTANIOUSLY, PRINTING THE QUERIES AS THEY ARE            *
*                RECEIVED. OUTGOING RESPONSE FORMAT IS ESTABLISHED THROUGH   *
*                THE USE OF A TEMPLATE FILE.                                 *
*                                                                            *
*                THIS IS A CLASSIC EXAMPLE OF HOW APPLICATIONS WORKING AS    *
*                PORTICO REMOTE HOST WOULD INTERFACE TO THE SUMMA FOUR      *
*                SEPT PRODUCT LINE.                                          *
*                                                                            *
*****************************************************************************
*                                                                            *
*                     RESTRICTED RIGHTS LEGEND                               *
```

```
* Use, duplication, or disclosure by Government Is Subject to restrictions as  *
* set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and    *
* Computer Software clause at DFARS 252.227-7013                              *
*                                                                             *
*****************************************************************************/



#define     ANSI
#define     ERROR   -1
#define     ALWAYS  1
#define     TRUE    1
#define     FALSE   0

#include    <stdio.h>
#include    <sys/types.h>
#include    <sys/limits.h>
#include    <sys/ipc.h>
#include    <sys/msg.h>
#include    <sys/sem.h>
#include    <sys/signal.h>

#include    <sys/socket.h>
#include    <netinet/in.h>
#include    <arpa/inet.h>
#include    <netdb.h>

#include    <api.h>
#include    <scpa.h>
#include    <sccl.h>
#include    <scprims.h>
#include    <sccp_macros.h>
#include    <tcap.h>

#include    <iu.h>


#define     TCTEMPLATE1     0
#define     STRING          "9876543210"


typedef struct sockaddr     SOCKADDR;
typedef struct sockaddr_in  SOCKADDR_IN;
typedef struct hostent      HOSTENT;

extern int  errno;
HOSTENT     *HostByNm;
u_short     IuPort;
int         SockFd, MsgSz;
char        XmtMsgBfr[sizeof (IUMSGX) + sizeof (TCAPAN) + sizeof (STRING)];
char        RcvMsgBfr[sizeof (IUMSGX) + sizeof (TCAPAN) + sizeof (STRING)];
IUMSGX      *XmtMsgX, *RcvMsgX;
TCAPANTEMP  *XmtTcTempMsg, *RcvTcTempMsg;
TCAPAN      *RcvTcMsg;



main (argc, argv)
int     argc;
char    **argv;
{
    SOCKADDR_IN     IuAdr;
```

```
void            cleanup(), (*signal())();
int             rcount;
HOSTENT         *gethostbyname();

            /* SEE THAT INCOMING ARGUMENT COUNT IS CORRECT. */


if (argc < 3)
{   printf ("USAGE: respcl [ServerName] [ServerPort]\n");
    exit (0);
}
            /* SETUP TO RECEIVE TERMINATION SIGNALS FROM THE
               PARENT. */

if (CatchSigs() == ERROR)
{   printf ("invoke: main()- Failed trying to setup SIGNALS...");
    exit (1);
}
            /* GET THE HOST BY IT'S NAME AND SETUP THE SOCKET
               STRUCTURE FOR THE SERVER PROCESS. */

if ((HostByNm = gethostbyname (argv[1])) == 0)
    return (ERROR);

IuPort = atoi (argv[2]);
memset((char *)&IuAdr, 0, sizeof(SOCKADDR_IN));
bcopy(HostByNm->h_addr, (char*)&IuAdr.sin_addr, HostByNm->h_length);
IuAdr.sin_family = AF_INET;
IuAdr.sin_port = htons (IuPort);

            /* CREATE A TCP/IP SOCKET. */

if ((SockFd = socket (AF_INET, SOCK_STREAM, 0)) < 0)
{   printf ("invoke: main()- Couldn't OPEN socket...");
    return (ERROR);
}
printf ("invoke: main()- Socket created... \n");

            /* CONNECT TO THE TCP SERVER. */

if (connect (SockFd, (SOCKADDR *)&IuAdr, sizeof (IuAdr)) < 0)
{   printf ("invoke: main- FAILED on CONNECT...\n");
    exit (0);
}
printf ("respcl: main()- Connected...\n");

            /* SETUP POINTER FOR THE TRANSMIT AND RECEIVED BUFFERS
               WHICH HOLD OUTGOING AND INCOMING SEPT MESSAGES. */

XmtMsgX = (IUMSGX*)XmtMsgBfr;
XmtTcTempMsg = (TCAPANTEMP*)XmtMsgX->msg;
RcvMsgX = (IUMSGX*)RcvMsgBfr;
RcvTcTempMsg = (TCAPANTEMP*)RcvMsgX->msg;
RcvTcMsg= (TCAPAN*)RcvMsgX->msg;

            /* INTIALIZE THE OUTGOING SEPT MESSAGE HEADER. */

XmtMsgX->HdrId = htons (GENERIC_SS7IU_HDR);
XmtMsgX->SessionId = 0;
XmtMsgX->state = htons (TRUE);
XmtMsgX->Ss7AdrFlag = htons (FALSE);
XmtMsgX->MsgFlavor = htons (IU_TCAP_ANSI_TEMP);
XmtMsgX->MsgLen = htons (sizeof (TCAPANTEMP) + sizeof (STRING) - 1);
```

```
            /* INITIALIZE THE SEPT ANSI TCAP TEMPLATE USED FOR OUTGOING
               SEPT SIGNALING. */

 XmtTcTempMsg->TempId = htons (TCTEMPLATE1);
 XmtTcTempMsg->DlgId = 0;

 XmtTcTempMsg->LinkEdIdFlg = htons (TRUE);
 XmtTcTempMsg->LinkEdId = 1;
 XmtTcTempMsg->InvokeIdFlg = htons (FALSE);
 XmtTcTempMsg->InvokeId = 0;

 XmtTcTempMsg->ParamFlag = htons (TRUE);
 XmtTcTempMsg->ParamLength = htons (sizeof (STRING) - 1);
 strncpy (XmtTcTempMsg->parameters, STRING, sizeof (STRING));
MsgSz = sizeof (IUMSGX) + sizeof (TCAPANTEMP) + ntohs (XmtTcTempMsg->ParamLength);

 while (ALWAYS)
 {
            /* ALWAYS READ THE FIXED LENGTH SEPT MESSAGE HEADER FIRST.
               IT WILL TELL YOU THE SEPT SS7 PROTOCOL FLAVOR THAT
               FOLLOWS AS WELL AS IT'S LENGTH. */

    if ((rcount = readn (SockFd, RcvMsgBfr, sizeof(IUMSGX))) == ERROR)
    {   printf ("main()- FAILURE on read of socket...");
        continue;
    }
    else if (rcount == 0)
    {   printf ("invoke: main()- TCP connection has been terminated by server.
Exiting...\n");
        cleanup();
        exit (1);
    }
            /* READ THE VARIABLE PROTOCOL FLAVOR PORTION OF THE SEPT
               MESSAGE AND PRINT IT. */

   if ((rcount = readn (SockFd, RcvMsgBfr + sizeof (IUMSGX), ntohs (RcvMsgX->Ms-
gLen))) == ERROR)
   {  fprintf (stderr, "invoke: main()- FAILURE on read of socket...\n");
        continue;
    }
    else if (rcount == 0)
    {   printf ("invoke: main()2- TCP connection has been terminated by server.
Exiting...\n");
        cleanup();
        exit (1);
    }

            /* PRINT THE CONTENTS OF THE INCOMING QUERY. */

    /*printf ("respcl: main()- Query received...\n");*/
    /*IuMsgPrint (RcvMsgX);*/

            /* COPY THE DIALOGUE ID AND INVOKE ID NUMBERS FROM THE QUERY INTO
               THE RESPONSE. */

    switch (RcvMsgX->MsgFlavor)
    {   case IU_TCAP_ANSI:
            XmtTcTempMsg->DlgId = RcvTcMsg->DlgId;
            XmtTcTempMsg->LinkEdIdFlg= RcvTcMsg->InvokeIdFlg;
            XmtTcTempMsg->LinkEdId = RcvTcMsg->InvokeId;
            break;
        case IU_TCAP_ANSI_TEMP:
```

```
                XmtTcTempMsg->DlgId = RcvTcTempMsg->DlgId;
                XmtTcTempMsg->LinkEdIdFlg= RcvTcMsg->InvokeIdFlg;
                XmtTcTempMsg->LinkEdId = RcvTcMsg->InvokeId;
                break;
            case IU_SCCP_ANSI_MNT:
                printf (“respcl: main()- SCCP message encountered. No response
   generated...\n”);
                continue;
            default:
              printf (“respcl: main()- Unidentified message encountered. No response
   generated...\n”);
                continue;
        }
                /* RESPOND TO THE INCOMING QUERY BY WRITING BACK A TCAP TEMPLATE
                   RESPONSE. */

      if ((rcount = writen (SockFd, XmtMsgBfr, sizeof (IUMSGX) + ntohs (XmtMsgX->Ms-
   gLen))) == ERROR)
           printf (“main()- FAILURE on write of socket...”);
    }
}
```

Appendix A
UNIX/VI EDITOR BASICS

A.1 UNIX BASICS

Table A.1: UNIX Basics

Command	Function
man **<command>**	manual pages describing command (detailed information about specified command)
date	brings up time and date
exit	exits session
who	shows who is on the system
env	displays environmental variables
more **<filename>**	displays text one page at a time
pwd	Print Working Directory
ls	LiSt files in directory
ls -aF	directories end with/ and hidden files
ls -l	long listing with added details
ls ~	list the contents of the home directory
ls -lrt	list long (detailed) names with latest date last
rm	remove (delete)
rmdir	remove directory
chmod 777 file.abc	change permissions for a file to all **+r**, **+w**, **+x**
chown owner **<filename>**	Change Owner of file or directory (must own that file)
cp file1 file2	copy file 1 to file 2
cp ~/file.ext .	copies from file.ext from server home directory to current location
tar cvf **<tarfile> <filenames>**	creates tarfile from filenames
tar xvf **<tarfile>**	eXtracts files from tarfile
tar tvf **<tarfile>**	output a Table of contents of tarfile
ps -ef	list all processes fully
su	Switch User (to **root**)

Table A.1: UNIX Basics (Continued)

Command	Function
tail **<filename>**	show file from the TAIL end
tail -f **<file>**	show new entries (i.e. tail -f cktint-Apr24.log)
fsck	File Systems cheCK
df -k	Disk Free (in Kilobytes) - shows disk utilization
du -k	Disk Utilization - shows space used by files in current directory, and those below (in Kilobytes)
jobs	show jobs running in background
fg **<number>**	brings job **<number>** to foreground
ctrl z	stops current process
bg	places stopped process in the background
kill **<pid>**	kills specified process (see ps -ef above)
Press “break” key (often F5), then, depending on prompt: >b -s **[or]** ok boot -s	Boots system into single user mode
vi **<file>**	enter **vi editor**

A.2 VI EDITOR

Table A.2: vi editor Commands

Command	Function
ESC	escape back to command mode
i	insert at cursor
a	append after cursor
x	delete under cursor
r	replace character under cursor
R	replace (typeover) a line
dw	delete word
dd	delete line
4dd	delete 4 lines
Y or yy	yank line (copy)
5Y or yy	yank a copy of 5 lines
p	put line (paste) below
P	put line above current line
u	undo last change
o	open line below
O	open line above
G	go to end of file
1G	go to first line of file
ctrl f	page forward
ctrl b	page backward
:q!	quit, do not save
:wq	write and quit
:sh	escape to shell (exit to return to **vi**)
:!**<command>**	execute UNIX command

Appendix B
FILE STRUCTURE

B.1 CKTINT CONFIGURATION

Figure B.1 is a illustration of the directory structure for the cktint configuration.

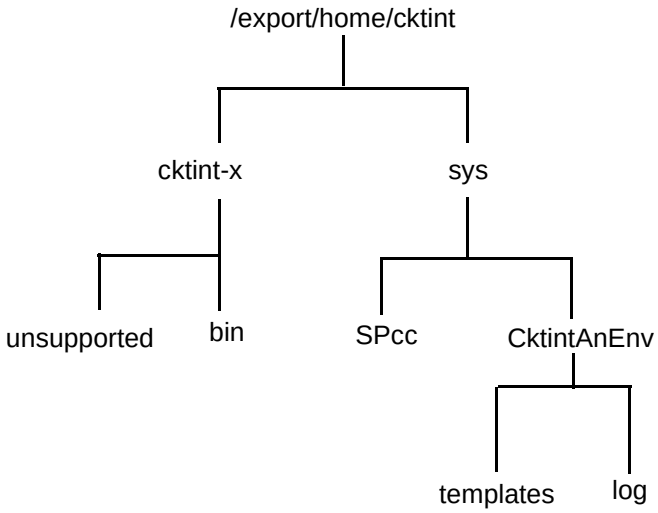


Figure B.1: Cktint Directory Tree

Table B.1: Cktint Directories

UNIX Directory Path	Shortcut	Contents
/export/home/cktint	$HOME	Subdirectories for the product
/export/home/cktint/cktint-x/bin	$BIN	Startup scripts
/export/home/cktint/cktint-x/ unsupported	$UN	The host emulator tool
/export/home/cktint/sys/CktintAnEnv	$XNV	All the executables and the configuration files used by cktint
/export/home/cktint/sys/CktintAnEnv/ templates	$XNV/ templates	The template files used by the cktint
/export/home/cktint/sys/CktintAnEnv/log	$XNV/log	The log directory
/export/home/cktint/sys/SPcc	$SPC $SPC1...7	.mml configuration files for ITU

B.2 SEPT CONFIGURATION

Figure B.1 is a illustration of the directory structure for SEPT configuration.

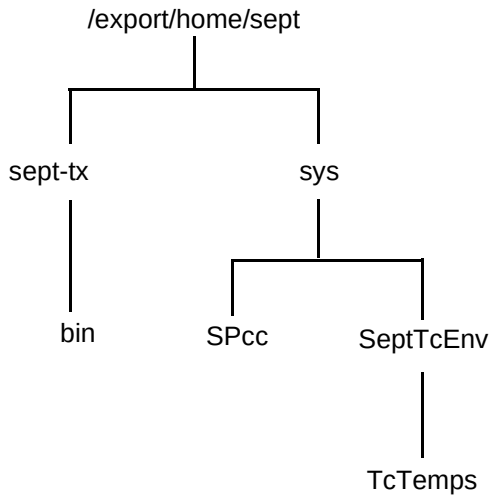


Figure B.2: SEPT Directory Tree

Table B.2: SEPT Directories

UNIX Directory Path	Shortcut	Contents
/export/home/sept	$HOME	Subdirectories for the product
/export/home/sept/sept-tx/bin	$BIN	Startup scripts
/export/home/sept/sys/SeptTcEnv	$SNV	All the executables and the configuration files used by sept
/export/home/sept/sys/SeptTcEnv/ TcTemps	$SNV/ TcTemps	The template files used by SEPT
/export/home/sept/sys/SPcc	$SPC $SPC1...7	.mml configuration files for ITU

Appendix C
UPGRADING/RE-INSTALLING THE OPERATING PLATFORM

C.1 INTRODUCTION

The Solaris Operating System is loaded on the SS7 subsystem at the factory. The information in this appendix is supplied in the event you need to upgrade or re-install the software.

C.2 PREPARING FOR UPGRADE/RE-INSTALLATION

1. If you have a redundant system, upgrade/install the Standby (SBY) side first.
2. Put the switch in the "A-ACT/B-SBY" state by running the Switch Active Side to Standby utility from the VCO/4K System Administration Console on the active side of the system. (Refer to the *VCO/4K System Administrator's Guide* for information on the utility.)
3. Hard-select the ACT side.
4. Move the VCO/4K AAC card's "Select Switch" from the "auto" position to the "A-ACT side."
5. Move the SS7 Fallback Switch "Select Switch" from the "auto" position to the "A-ACT side."

C.3 RELOADING THE SOLARIS OPERATING SYSTEM

1. Log in to the SS7 subsystem as **cktint**.
2. On the standby side, enter the following command and press **Return**:

   stop-ss7.sh

   Type **px** and press **Return** to verify no processes are running.
3. Create a temporary directory on the SS7 A-Active side or on any other machine and FTP/copy the following from B-side (standby):
   - **EBSkeyfile.dat** file from B-side (located in **/export/home** or in **$EBSHOME/access/dat**)
   - **$SPC/mml** files
   - **$XNV/ckt***
   - **grp***
   - **CktInt.cfg** files
   - **SS7 $XNV/template** files
   - **/etc/hosts file** (just to reference the hostnames and IP addresses)

4. Once you have saved the files from B-side (standby), you may continue reloading the Solaris Operating System.

   NOTE: Check the Solaris documentation for specific references.

5. Make sure you have the following items ready:
   - the CD ROM drive shipped with the system
   - the power cord for the CD ROM drive
   - the SCSI cable for the CD ROM drive
   - the Solaris OS CD ROM shipped with the system
6. Put the CD ROM drive near the Sparc on a flat surface.
7. Connect the power and SCSI cables.
8. Insert the Solaris OS CD ROM into the drive.
9. Make sure you have a console attached.
10. Log in to the SS7 subsystem as **root**.
11. Type **halt** and press **Return**.
12. At O.K. prompt, enter the following command and press **Return**:

    ```
    boot cdrom - install
    ```

13. Once the system boots, it will come up as if you ran sys-config for the first time. Follow the directions in *Section 3.4* for running sys-config.

    NOTE: If your SS7 subsystem console is a VT100 terminal, press **PF2** *on the numeric keypad instead of the* **F2** *key.*

C.4 INSTALLING SOLARIS 2.6 PATCHES

1. If you haven't already done so, log in to the SS7 subsystem as **cktint**.
2. Stop the SS7 stack by entering the following command and pressing **Return**:

   ```
   stop-ss7.sh
   ```

   Type **px** and press **Return** to verify no processes are running.

3. Create a new directory named **sol2.6patches** under **/export/home** on the SS7 system.
4. **cd** to **sol2.6patches**.
5. From **%** prompt, FTP to **Cisco/S4** ftp server to get the required Solaris 2.6 patches.
6. Enter the following command and press **Return**:

   ```
   ftp 198.92.30.33
   ```

   user = ftp

   passwd = your full e-mail address

7. At the **ftp>** prompt, enter your user name and press **Return**.
8. At the **pwd** prompt, enter your full e-mail address and press **Return**.

9. The ftp server will respond with the following:

   "/" is the current directory

10. Enter the following command and press **Return**:

    cd /mhaider/sol2.6patch

11. Type **ls -l** and press **Return**.
12. You will find the following files under the specified directory:

```
-rw-r--r--  1 mhaider  eng     3327787 Jan 15  1999 105181-04.tar.Z
-rw-r--r--  1 mhaider  eng     2464611 Jan 15  1999 105210-06.tar.Z
-rw-r--r--  1 mhaider  eng      116549 Jan 15  1999 105216-02.tar.Z
-rw-r--r--  1 mhaider  eng      164807 Sep 29 11:56 105529-08.tar.Z
-rw-r--r--  1 mhaider  eng      186797 Jan 15  1999 105604-02.tar.Z
-rw-r--r--  1 mhaider  eng      117369 Jan 15  1999 105615-03.tar.Z
-rw-r--r--  1 mhaider  eng      108697 Jan 15  1999 105665-01.tar.Z
-rw-r--r--  1 mhaider  eng      124689 Jan 15  1999 105667-01.tar.Z
-rw-r--r--  1 mhaider  eng      216433 Jan 15  1999 105786-04.tar.Z
-rw-r--r--  1 mhaider  eng      136995 Jan 15  1999 106125-02.tar.Z
-rw-r--r--  1 mhaider  eng        1296 Jan 15  1999 patchinstall2.6
```

13. Set mode to binary by entering **bin** and pressing **Return**.
14. Type **hash** and press **Return**.
15. Type **mget *** and press **Return**.
16. To get each file type, for "yes" type **y** and press **Return**.
17. After you get all the files into directory **sol2.6patches**, type **bye** and press **Return** to quit the FTP session.
18. Change permissions on file **sol2.6patches** to **777**.
19. Switch user to **su**.
20. At the prompt, enter the following command and press **Return**:

    ./patchinstall2.6

    This will install Solaris 2.6 patches on your system.

    NOTE: If the CD ROM you re-installed from contained higher level patches, you will receive warning messages.

21. Reboot the SS7 subsystem.
22. Log in to the SS7 subsystem as **cktint**.

C.5 FINISHING UPGRADE/RE-INSTALLATION ON REDUNDANT SYSTEM

1. After the system comes up successfully on the B-side (SBY), put the switch in the "B-ACT/A-SBY" state by running the Switch Active Side to Standby utility from the VCO/4K System Administration Console on the active side of the system. (Refer to the *VCO/4K System Administrator's Guide* for information on the utility.)
2. Hard-select the ACT side.
3. Move the VCO/4K AAC card's "Select Switch" from the "auto" position to the "B-ACT side."
4. Move the SS7 Fallback Switch "Select Switch" from the "auto" position to the "B-ACT side."
5. Check to see if links are aligned, calls are being processed and the system is fully operational.
6. Repeat all the steps for reloading the Solaris Operating System on the A-side.

C.6 ERRORS AFTER UPGRADE/RE-INSTALLATION

After the upgrade/re-installation, you may find errors such as:

ld.so.1:/export/home/EBS/access/bin/AccessAlarm:fatal:libC.so.3.0:can't open

-or-

"Alarm Handler" errors

Should you find any errors on the SS7 startup, do the following:

1. Log in to the SS7 subsystem as **cktint**.
2. Enter the following command and press **Return**:

   stop-ss7.sh

   Type **px** and press **Return** to verify no processes are running.
3. **cd** to **/usr/lib**.
4. Type **ls -l libC*** and press **Return**.
5. The following files should display:

   libC.so.3

   libCso.5
6. Change user as root (super user) by typing **su** followed by password.

7. At **hash** prompt, enter the following command and press **Return**:

   ln -s libC.so.3 libC.so.3.0

   This will create a symbolic link between the two files.

8. Exit from **root**.

9. Enter the following command and press **Return**:

   start-ss7.sh

10. This should bring up the stack without any problems. You should also be able to see Access Alarms.

Appendix D
UPGRADING/RE-INSTALLING THE SS7 SUBSYSTEM SOFTWARE

D.1 INTRODUCTION

The software is loaded on the SS7 subsystem hard drive at the factory. The information in this appendix is supplied in the event you need to upgrade or re-install the software.

D.2 RE-INSTALLING THE SERIAL PORT SBUS CARD SOFTWARE

Reinstall the software for the modem's serial port on the Sbus card following the procedures in the manufacturer's documentation (Magma).

D.3 PREPARING FOR UPGRADE/RE-INSTALLATION

1. If you have a redundant system, upgrade/install the Standby (SBY) side first.
2. Put the switch in the "A-ACT/B-SBY" state by running the Switch Active Side to Standby utility from the VCO/4K System Administration Console on the active side of the system. (Refer to the *VCO/4K System Administrator's Guide* for information on the utility.)
3. Hard-select the ACT side.
4. Move the VCO AAC card's "Select Switch" from the "auto" position to the "A-ACT side."
5. Move the SS7 Fallback Switch "Select Switch" from the "auto" position to the "A-ACT side."
6. Log in to the SS7 subsystem as **cktint**.
7. On the standby side, enter the following command and press **Return**:

   stop-ss7.sh

   Type **px** and press **Return** to verify no processes are running.

8. Create a temporary directory on the SS7 A-Active side or on any other machine and FTP/copy the following from B-side (standby):
   - **EBSkeyfile.dat** file from B-side (located in **/export/home** or in **/export/home/EBS/access/dat**)
   - **/export/home/cktint/sys/SPcc<n>/mml** files
   - **/export/home/cktint/sys/CktintAnEnv/ckt*** files
   - **/export/home/cktint/sys/CktintAnEnv/grp*** files
   - **/export/home/cktint/sys/CktintAnEnv/CktInt.cfg** file
   - **/export/home/cktint/sys/CktintAnEnv/template/*** files
   - **/etc/hosts** file (just to reference the hostnames and IP addresses)

   NOTE: Always use full path names when for FTP.

D.4 UPGRADING/RE-INSTALLING THE SS7 SUBSYSTEM SOFTWARE

This subsection describes how to upgrade/re-install all of the SS7 subsystem software.

D.4.1 Copying Software Diskettes, Changing File Permissions, and Removing EBS Drivers

1. If you haven't already done so, log in to the SS7 subsystem as **root**.
2. **cd** to **/export/home**.
3. Enter the following command and press **Return**:

   /etc/init.d/volmgt stop

4. Copy the SS7 subsystem floppies to the system drive:
   a. Insert the first EBS Software floppy in the SS7 subsystem drive.
   b. Enter the following command and press **Return:**

      cpio -icduv -C65536 -I/dev/rfd0

      When the system is finished copying disk 1, the following message appears:

      End of medium on "input"
      Change to part 2 and press RETURN key. [q]

   c. Insert the second floppy and press **Return**.
   d. Repeat Steps A through C for each software component (EBS software, Circuit Interworking, SEPT, and PPP) until all of the floppies are copied to the system disk.

   NOTE: You only have to copy the diskettes for the software component(s) you need.

5. Enter the following command and press **Return**:

   /etc/init.d/volmgt start

6. When the system has copied all of the SS7 software, change the permissions on the install scripts:
   a. Change permissions on the Circuit Interworking install script by entering the following command and pressing **Return**:

      chmod 755 ./install_cktint.sh

   b. Change permissions on the EBS install script by entering the following command and pressing **Return**:

      chmod 755 ./install_ebs.sh

7. **cd** to **$EBSHOME/access/install**
8. Type **./rmebsdrv** and press **Return**.
9. Type **y** at the prompt and press **Return.**
10. Type **reboot** and press **Return**.

D.4.2 Re-Installing Circuit Interworking.

1. Log in to the SS7 subsystem as **root**.
2. Enter the following command and press **Return**:

   ./install_cktint.sh

   Messages similar to the following appear:

   ```
   ##########################################
   Installing cktint (File: cktint.cpio.Z)
   ##########################################
   ```

   *NOTE: If the directory **/export/home/cktint** already exists, you will be asked if you would like to move it with the following message:*

   ```
   The directory /export/home/cktint already exists.
   Would you like to move /export/home/cktint [y/n]?
   ```

   If you do not move cktint to a new name, the script will terminate.

   Type **y** *and press* **Return**.

   *Then, enter a directory extension name. This will move the existing **/export/home/cktint** to **/export/home/cktint.old**. Continue with the installation.*

   ```
   Creating cktint account...
   done
   Extracting files...
   done
   Setting up...
   done
   ```

*NOTE: If **/etc/rc3.d/S85ss7** does not exist, you will be asked if you want to install the autostart option with the following message:*

```
Autostart script S85ss7 does not exist.
Would you like to install it [y/n]?
```

Type **y** *and press* **Return**. *The following message appears:*

```
Installing /etc/rc3.d/S85ss7...done


Adding entries to /etc/system file
Adding tcp timers to /etc/rc2.d/S69inet file
done

Before using the SS7 subsystem, please reboot.
```

*NOTE: **Do not** reboot yet.*

Rename this file by entering the following and pressing Return:

```
mv/etc/rc3d/S85ss7/etc/rc3d/s85temp
```

-OR-

Type **n** *and press* **Return**. *Autostart is **not** installed.*

3. **cd** to **$EBSHOME/access/install**
4. Type **./setupebsdrv** and press **Return**.
5. Type **y** at the prompt and press **Return.**
6. The installation script for the Circuit Interworking software creates the user ID **cktint**. Use this user ID when performing Circuit Interworking software functions as described in this section and *Section 3*.
7. When the installation script is complete, the UNIX command prompt appears.

At this point, the SS7 subsystem software is configured with the user ID **cktint**. The home directory for **cktint** is **/export/home/cktint ($HOME)**.

D.4.3 Re-installing EBS

1. Log in to the SS7 subsystem as **root**.
2. Change permissions on the **.cshrc** script:
   a. **cd** to **/export/home/cktint**.
   b. Change permissions on the **.cshrc** script by entering the following command and pressing **Return**:

      chmod 644 .cshrc
3. **cd** to **/export/home**.
4. At the system prompt, enter the following command and press **Return**:

   ./install_ebs.sh

   Messages similar to the following appear:

   ```
   ##########################################
   Installing EBS Software (File: ebs.CC34.00.sbus.cpio.Z)

   ##########################################

   Extracting files from archive. This will take a little while ....
   The Solaris kernel drv and strmod modules will now be rebuilt with the EBS Software.
   EBS home is /export/home/EBS
   Calling rmEBSdrv in /export/home/EBS/access/install


   You are about to remove the AccessMANAGER modules.
   Do you wish to continue[y/n]?
   ```
5. Type **y** and press **Return**. Messages similar to the following appear:

   ```
   Unloading ANTC module ...

   ANTC module is now unloaded.

   .

   .

   All AccessMANAGER modules are now removed.


   Calling setupEBSdrv in /export/home/EBS/access/install


   You are about to install the AccessMANAGER modules.
   Do you wish to continue [y/n]?
   ```

6. Type **y** and press **Return**. Messages similar to the following appear:

   Installing SPM module ...
   SPM module is now installed.

   .

   .

   Installation of AccessMANAGER modules is now complete.

   The new Solaris drv and strmod modules are now installed.

   Installation complete. Reboot not required.

7. To configure PPP, complete the following steps:
   a. Run the **install_ppp.sh** script.
   b. Acquire two more IP addresses from your system administrator. Ensure that the addresses are on the same network as the SS7 subsystem.
   c. Verify that your user ID is still root and open the **/etc/hosts** file in located in the system's root directory.
   d. Modify the **/etc/hosts** file to include the two new lines:

      <ipaddress><hostname>ppp
      <ipaddress>ppp-dialinpc

      where the two <ipaddress> variables are the IP Addresses you acquired from your system administrator in Step 9b and <hostname> is the name of the SS7 subsystem.

   NOTE: If the SS7 subsystem is part of the greater network that is running NIS or NIS+, the /etc/hosts file of the name server's must be modified to include these changes as well.

   e. Modify the **/etc/asppp.cf** file to include the following line before the "defaults" line.

      ifconfig ipdptp0 plumb <hostname>ppp down

      where <hostname> is the name of the SS7 subsystem.
   f. Modify the permissions on the **/etc/asppp.cf** file using the following command:

      chmod 600 /etc/asppp.cf

8. Apply all modifications to the appropriate configuration files for your specific system.
9. Reboot the SS7 subsystem.

D.4.4 Restoring Configuration

1. Log in to the SS7 subsystem as **cktint**.
2. FTP and recover all configuration files that were preserved/saved from the B-side prior to beginning reload and place them in their correct location.
3. Once the files are placed/copied to their proper location, you can start rebuilding the database configuration.
4. **cd** to **$EBSHOME/access/RUN0/DBfiles**.
5. Remove all files, if any, by typing **rm <filename>** and pressing **Return**.
6. Type **cd** and press **Return**.
7. Type **cd backup** and press **Return**.
8. Remove all files, if any, by typing **rm <filename>** and pressing **Return**.
9. Repeat step 4 through step 8 for each additional SP, replacing **RUN0** with **RUN<n>**.
10. Type **cd $SPC** (each directory after SP 0 is **SPcc<n>**) and press **Return**.
11. In the mtp.mml file, change/verify your port numbers to 1, 2, 3, 4 (5, 6, 7, 8, depending on your configuration) if you are using all links/ports.

Now you have removed the old database and want to configure the new. The following steps/procedures are always performed when any changes are made to the .mml files:

1. On prompt, enter the following command and press **Return**:

   ebs_start

   Wait for some time and press **Return**.
2. On prompt, enter the following command and press **Return**:

   upmd 0 &

   Wait for some time and press **Return**.
3. On prompt, enter the following command and press **Return**:

   snmd 0 &

   Wait for some time and press **Return**.
4. Make sure you are in directory **$SPC**.
5. On prompt, enter the following command and press **Return**:

   mml 0 mtp.mml

   You should see successes for all lines for the **mtp.mml** file.
6. On prompt (if using TCAP), enter the following command and press **Return**:

   scmd 0 &

   Wait for some time and press **Return**.

7. On prompt (if using TCAP), enter the following command and press **Return**:

   mml 0 tcap.mml

   You should see successes for all lines for the **tcap.mml** file.

8. On prompt, enter the following command and press **Return**:

   AccessISUP 0 &

9. On prompt, enter the following command and press **Return**:

   mml 0 isup.mml

   You should see successes for all lines for the **isup.mml** file.

10. Repeat step 2 through step 9 for each SP, replacing the 0 with the appropriate SP number (i.e. upmd 1-7 &, snmd 1-7 &, mml 1-7 mtp.mml, scmd 1-7 &, mml 1-7 tcap.mml, AccessISUP 1-7 &, and mml 1-7 isup.mml).

11. Enter the following command and press **Return**:

    stop-ss7.sh

    Type **px** and press **Return** to verify no processes are running.

12. If redundant, **cd** to /**export/home/cktint**.

13. Enter the following command and press **Return**:

    chmod 644 $HOME/.cshrc

14. Use the vi editor to open the file **$HOME/.cshrc** and perform the following steps:

    a. Remove the comment symbol (#) from the following line:

       `#setenv PLTFRMTYP REDUNDANT`

    b. Add the comment symbol (#) to the following line:

       `setenv PLTFRMTYP STANDALONE`

    c. Save the changes and close the file.

15. To have the change take affect (without logging out and logging back in again), enter the following command and press **Return**.

    source $HOME/.cshrc

16. Use the vi editor to open the file **$EBSHOME/access/config/AccessRd.cfg**.
    a. Check the following lines. The lines should read:

        MONITOR_OPTION OFF

        CTS_CONFIGURATION INVERTED

        HOST-A <SS7 subsystem side A hostname>

        (i.e., HOST-A tsup6ss7a)

        HOST-B <SS7 subsystem side B hostname>

        (i.e., HOST-B tsup6ss7b)

    b. If the lines do not match, modify them accordingly.
    c. Save the changes and close the file.
17. Log on as **root** to the SS7 subsystem side B, **<hostname>b**, and repeat step 13 through step 16.
18. Check to make sure the SS7 Integrated Software version matches that on side A. Enter the following commands and press **Return** after each:

        %cd $XNV

        %version cktint

19. Enter the following command and press **Return**:

        start-ss7.sh

    Type **px** and press **Return** to verify no processes are running.

D.5 RE-INSTALLING THE TCAP SS7 SUBSYSTEM SOFTWARE

To re-install TCAP, complete the following steps:

1. If you haven't already done so, log in to the SS7 subsystem as **root**.
2. **cd** to **/export/home**.
3. Enter the following command and press **Return**:

   /etc/init.d/volmgt stop

4. Copy the SS7 subsystem TCAP SEPT floppy to the system drive:
   a. Insert the SEPT floppy in the SS7 subsystem drive.
   b. Enter the following command and press **Return:**

      cpio -icduv -C65536 -I/dev/rfd0

5. Enter the following command and press **Return**:

   /etc/init.d/volmgt start

6. When the system has copied the software, change the permissions on the SEPT install script by entering the following command and pressing **Return**:

   chmod 755 ./install_sept.sh

7. Enter the following command and press **Return**:

   ./install_sept.sh

   If a **sept** account does not exist, the following messages appear:

   Installing sept (File: 'basename sept.cpio.Z)

   Creating sept password entry...

   Creating sept shadow entry...

   done

   Change password now...

8. Enter a password and press **Return**.

   When the SEPT software is completely installed, the following messages appear:

   Extracting files...

   done

   Setting up...

   done

9. Remove the SEPT diskette.

D.6 RE-INSTALLING THE EBS SS7 LICENSE FILE

To reinstall the EBS SS7 license file, complete the following steps:

1. If you haven't already done so, log in as **root**.
2. Insert the EBS SS7 license floppy diskette into the floppy drive.
3. Change directory (**cd**) to **/export/home**.
4. Enter the following command and press **Return**:

   /etc/init.d/volmgt stop

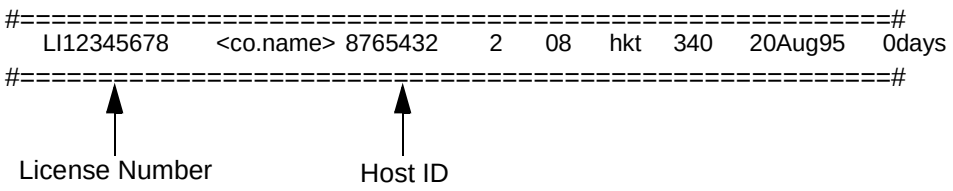
5. Enter the following command and press **Return**:

   tar xvf /dev/rfd0

6. Enter the following command and press **Return**:

   /etc/init.d/volmgt start

7. Verify that you have a valid license by completing the following steps:
   a. Execute the UNIX **hostid** command and note the system's Host ID.
   b. Use the vi editor to open the **EBSkeyfile.dat** file. This file contains a line that looks similar to the following:



   The Host ID listed in this file should match the system's Host ID.

   c. Exit the vi Editor. If the Host IDs do not match, the license is invalid and the EBS SS7 software will not run. Contact Cisco Systems Technical Support.

Appendix E
ISUP MESSAGE TYPES AND PARAMETERS

E.1 ISUP MESSAGE TYPES

Table E.1: ISUP Message Types Listed By Binary Code

Hex Code	Binary Code	Message Type	Acronym
01	0000 0001	Initial Address	IAM
02	0000 0010	Subsequent Address	SAM
03	0000 0011	Information Request	INR
04	0000 0100	Information	INF
05	0000 0101	Continuity	COT
06	0000 0110	Address Complete	ACM
07	0000 0111	Connect	CON
08	0000 1000	Forward Transfer	FOT
08	0000 1000	Forward Transfer	FOT
09	0000 1001	Answer	ANM
0A	0000 1010	Reserved	–
0B	0000 1011	Reserved	–
0C	0000 1100	Release	REL
0D	0000 1101	Suspend	SUS
0E	0000 1110	Resume	RES
0F	0000 1111	Reserved	–
10	0001 0000	Release Complete	RLC
11	0001 0001	Continuity Check Request	CCR
12	0001 0010	Reset Circuit	RSC
13	0001 0011	Blocking	BLO
14	0001 0100	Unblocking	UBL
15	0001 0101	Blocking Acknowledgment	BLA
16	0001 0110	Unblocking Acknowledgment	UBA

Table E.1: ISUP Message Types Listed By Binary Code (Continued)

Hex Code	Binary Code	Message Type	Acronym
17	0001 0111	Circuit Group Reset	GRS
18	0001 1000	Circuit Group Blocking	CGB
19	0001 1001	Circuit Group Unblocking	CGU
1A	0001 1010	Circuit Group Blocking Acknowledgment	CGBA
1B	0001 1011	Circuit Group Unblocking Acknowledgment	CGUA
1C	0001 1100	Call Modification Request	CMR
1D	0001 1101	Call Modification Completed	CMC
1E	0001 1110	Call Modification Reject	CMRJ
1F	0001 1111	Facility Request	FAR
20	0010 0000	Facility Accepted	FAA
21	0010 0001	Facility Reject	FRJ
27	0010 0111	Delayed Release	DRS
28	0010 1000	Pass Along Message	PAM
29	0010 1001	Circuit Group Reset Acknowledgment	GRA
2A	0010 1010	Circuit Query	CQM
2B	0010 1011	Circuit Query Response	CQR
2C	0010 1100	Call Progress	CPG
2D	0010 1101	User-to-User Information	USR
2E	0010 1110	Unequipped Circuit Identification Code	UCIC
2F	0010 1111	Confusion	CFN
30	0011 0000	Overload	OLM
31	0011 0001	Charge Information	CRG
32	0011 0010	Network Resource Management	NRM
36	0011 0110	Identification Request	IDR
37	0011 0111	Identification Response	IRS
38	0011 1000	Segmentation	SGM

Table E.2: ISUP Message Types Listed By Name

Message Type	Acronym	Hex Code	Binary Code
Address Complete	ACM	06	0000 0110
Answer	ANM	09	0000 1001
Blocking	BLO	13	0001 0011
Blocking Acknowledgment	BLA	15	0001 0101
Call Modification Completed	CMC	1D	0001 1101
Call Modification Reject	CMRJ	1E	0001 1110
Call Modification Request	CMR	1C	0001 1100
Call Progress	CPG	2C	0010 1100
Charge Information	CRG	31	0011 0001
Circuit Group Blocking	CGB	18	0001 1000
Circuit Group Blocking Acknowledgment	CGBA	1A	0001 1010
Circuit Group Reset	GRS	17	0001 0111
Circuit Group Reset Acknowledgment	GRA	29	0010 1001
Circuit Group Unblocking	CGU	19	0001 1001
Circuit Group Unblocking Acknowledgment	CGUA	1B	0001 1011
Circuit Query	CQM	2A	0010 1010
Circuit Query Response	CQR	2B	0010 1011
Confusion	CFN	2F	0010 1111
Connect	CON	07	0000 0111
Continuity	COT	05	0000 0101
Continuity Check Request	CCR	11	0001 0001
Delayed Release	DRS	27	0010 0111
Facility Accepted	FAA	20	0010 0000
Facility Reject	FRJ	21	0010 0001
Facility Request	FAR	1F	0001 1111
Forward Transfer	FOT	08	0000 1000
Identification Request	IDR	36	0011 0110
Identification Response	IRS	37	0011 0111

Table E.2: ISUP Message Types Listed By Name (Continued)

Message Type	Acronym	Hex Code	Binary Code
Information	INF	04	0000 0100
Information Request	INR	03	0000 0011
Initial Address	IAM	01	0000 0001
Network Resource Management	NRM	32	0011 0010
Pass Along Message	PAM	28	0010 1000
Release	REL	0C	0000 1100
Release Complete	RLC	10	0001 0000
Reset Circuit	RSC	12	0001 0010
Reserved	–	0A	0000 1010
Reserved	–	0B	0000 1011
Reserved	–	0F	0000 1111
Resume	RES	OE	0000 1110
Segmentation	SGM	38	0011 1000
Subsequent Address	SAM	02	0000 0010
Suspend	SUS	0D	0000 1101
Unblocking	UBL	14	0001 0100
Unblocking Acknowledgment	UBA	16	0001 0110
Unequipped Circuit Identification Code	UCIC	2E	0010 1110
User-to-User Information	USR	2D	0010 1101

E.2 ISUP MESSAGE PARAMETERS

Table E.3: ISUP Message Parameters Listed By Binary Code

Hex Code	Binary Code	Parameter Name
00	0000 0000	End of Optional Parameters
01	0000 0001	Call Reference
02	0000 0010	Transmission Medium Requirements
03	0000 0011	Access Transport
04	0000 0100	Called Party Number

Table E.3: ISUP Message Parameters Listed By Binary Code (Continued)

Hex Code	Binary Code	Parameter Name
05	0000 0101	Subsequent Number
06	0000 0110	Nature of Connection Indicators
07	0000 0111	Forward Call Indicators
08	0000 1000	Optional Forward Call Indicators
09	0000 1001	Calling Party's Category
0A	0000 1010	Calling Party Number
0B	0000 1011	Redirecting Number
0C	0000 1100	Redirection Number
0D	0000 1101	Connection Request
0E	0000 1110	Information Request Indicators
0F	0000 1111	Information Indicators
10	0001 0000	Continuity Indicators
11	0001 0001	Backward Call Indicators
12	0001 0010	Cause Indicators
13	0001 0011	Redirection Information
14	0001 0100	Reserved ITU Codes
15	0001 0101	Circuit Group Supervision Message Type Indicator
16	0001 0110	Range and Status
17	0001 0111	Call Modification Indicators
18	0001 1000	Facility Indicator
1A	0001 1010	Closed User Group Interlock Code
1D	0001 1101	User Service Information
1E	0001 1110	Signaling Point Code
1F	0001 1111	Reserved ITU Codes
20	0010 0000	User-to-User Information
21	0010 0001	Connected Number
22	0010 0010	Suspend/Resume Indicators
23	0010 0011	Transit Network Selection
24	0010 0100	Event Information Indicators

Table E.3: ISUP Message Parameters Listed By Binary Code (Continued)

Hex Code	Binary Code	Parameter Name
26	0010 0110	Circuit State Indicator
27	0010 0111	Automatic Congestion Level
28	0010 1000	Original Called Number
29	0010 1001	Optional Backward Call Indicators
2A	0010 1010	User-to-User Indicators
2B	0010 1011	Origination ISC Point Code
2C	0010 1100	Generic Notification
2D	0010 1101	Call History Information
2E	0010 1110	Access Delivery Information
2F	0010 1111	Network Specific Facilities
30	0011 0000	User Service Information Used
31	0011 0001	Propogation Delay Counter
32	0011 0010	Remote Operations
33	0011 0011	Service Activation
34	0011 0100	User Teleservice Information
35	0011 0101	Transmission Medium Used
36	0011 0110	Call Diversion Information
37	0011 0111	Echo Control Information
38	0011 1100	Message Compatibility Information
39	0011 1001	Parameter Compatibility Information
3A	0011 1010	MLPP Precedence
3B	0011 1011	MCID Request Indicator
3C	0011 1100	MCID Response Indicator
3E	0011 1110	Transmission Medium Requirement Prime
3F	0011 1111	Location Number
40	0100 0000	Redirection Number Restriction
42	0100 0010	Generic Reference
C0	1100 0000	Generic Address
C1	1100 0001	Generic Digits

Table E.4: ISUP Message Parameters Listed By Name

Parameter Name	Hex Code	Binary Code
Access Delivery Information	2E	0010 1110
Access Transport	03	0000 0011
Automatic Congestion Level	27	0010 0111
Backward Call Indicators	11	0001 0001
Call Diversion Information	36	0011 0110
Call History Information	2D	0010 1101
Call Modification Indicators	17	0001 0111
Call Reference	01	0000 0001
Called Party Number	04	0000 0100
Calling Party Number	0A	0000 1010
Calling Party's Category	09	0000 1001
Cause Indicators	12	0001 0010
Circuit Group Supervision Message Type Indicator	15	0001 0101
Circuit State Indicator	26	0010 0110
Closed User Group Interlock Code	1A	0001 1010
Connected Number	21	0010 0001
Connection Request	0D	0000 1101
Continuity Indicators	10	0001 0000
Echo Control Information	37	0011 0111
End of Optional Parameters	00	0000 0000
Event Information Indicators	24	0010 0100
Facility Indicator	18	0001 1000
Forward Call Indicators	07	0000 0111
Generic Address	C0	1100 0000
Generic Digits	C1	1100 0001
Generic Notification	2C	0010 1100
Generic Reference	42	0100 0010
Information Indicators	0F	0000 1111

Table E.4: ISUP Message Parameters Listed By Name (Continued)

Parameter Name	Hex Code	Binary Code
Information Request Indicators	0E	0000 1110
Location Number	3F	0011 1111
MCID Request Indicator	3B	0011 1011
MCID Response Indicator	3C	0011 1100
Message Compatibility Information	38	0011 1000
MLPP Precedence	3A	0011 1010
Nature of Connection Indicators	06	0000 0110
Network Specific Facilities	2F	0010 1111
Optional Backward Call Indicators	29	0010 1001
Optional Forward Call Indicators	08	0000 1000
Original Called Number	28	0010 1000
Origination ISC Point Code	2B	0010 1011
Parameter Compatibility Information	39	0011 1001
Propogation Delay Counter	31	0011 0001
Range and Status	16	0001 0110
Redirecting Number	0B	0000 1011
Redirection Information	13	0001 0011
Redirection Number	0C	0000 1100
Redirection Number Restriction	40	0100 0000
Remote Operations	32	0011 0010
Reserved ITU Codes	14 1F	0001 0100 0001 1111
Service Activation	33	0011 0011
Signaling Point Code	1E	0001 1110
Subsequent Number	05	0000 0101
Suspend/Resume Indicators	22	0010 0010
Transit Network Selection	23	0010 0011
Transmission Medium Requirement Prime	3E	0011 1110
Transmission Medium Requirements	02	0000 0010

Table E.4: ISUP Message Parameters Listed By Name (Continued)

Parameter Name	Hex Code	Binary Code
Transmission Medium Used	35	0011 0101
User Service Information	1D	0001 1101
User Service Information Prime	30	0011 0000
User Teleservice Information	34	0011 0100
User-to-User Indicators	2A	0010 1010
User-to-User Information	20	0010 0000

Appendix F
COUNTRY VARIANTS

This appendix lists the variants for Hong Kong, Thailand, Chile, Singapore, Germany, Switzerland, Italy, Finland, Australia, and Spain.

F.1 HONG KONG

Table F.1: Hong Kong Variant

ITU standards supported by Hong Kong	Q.701-Q.703, 1988 (Blue Book) Q.704-Q.707, 1988 (Blue Book) Q.711-Q.714, 1988 (Blue Book) Q.761-Q.764, 1988 (Blue Book)
Hong Kong specific messages added	Delayed Release (DRS) Unequipped Circuit (UCIC)
Unsupported messages	Subsequent Address (SAM) Forward Transfer (FOT) Call Modification Completed (CMC) Call Modification Reject (CMJ) Call Modification Request (CMR) Facility Reject (FCJ) Facility Request (FCR) Facility Accept (FCA) Continuity Check Request (CCR) Pass Along (PSA) User-To-User Information (UUI) Overload (OVL) CHI (Charge Information)

Table F.1: Hong Kong Variant(Continued)

Unsupported parameters	Call Reference (CRF) Access Transport (AXP) Subsequent Number (SAN) Optional Forward Call Indicators (OFI) Redirection Number (REDI) Connection Request (CONF) Call Modification Indicator (CMI) Facility Indicator (FCI) Closed User Group Interlock Code (CUGC) User Service Information (USI) Signaling Point Code (SPC) User-To-User Information (UUIN) Connected Number (CONN) Transit Network Selection (XNS) Automatic Congestion Level (ACL) User-To-User Indicator (UUIC)

F.2 THAILAND

Table F.2: Thailand Variant

ITU standards supported by Thailand	Q.701-Q.703, 1988 (Blue Book) Q.704-Q.707, 1988 (Blue Book) Q.711-Q.714, 1988 (Blue Book) Q.761-Q.764, 1988 (Blue Book)
Thailand specific messages added	Offering Message (OFR) False Answering Message (FLA)
Thailand specific parameters	Cause Value #102 is used in the RELEASE message when timers T2 or T6 expire. Refer to the *Specification for Common Signaling No. 7 of Telephone Organizations of Thailand, Version 1.2* (September, 1993) for more information on cause values and timers.
Thailand specific timers added	Timer T2

F.3 CHILE

Table F.3: Chile Variant

ITU standards supported by Chile	Q.701-Q.703, 1988 (Blue Book) Q.704-Q.707, 1988 (Blue Book) Q.711-Q.714, 1988 (Blue Book) Q.761-Q.764, 1988 (Blue Book) Q.767, 1988 (Blue Book)
Chile specific messages added	Messages are based on the ITU Q.767 standards. In addition, the following messages from the ITU Q.764 standards are supported: Information (INF) Information Request (INR)
Chile specific parameters added	Parameters are based on the ITU Q.767 standards, but certain parameters are only used for specific events. Cause Indicators in CONFUSION (CFN) and RELEASE (REL) messages that are generated by the ISUP layer have a Cause Value of “Protocol error-unspecified.” Cause Indicator for the REL message that is generated by the ISUP layer because the T6 timer has expired has a Cause Value of “Normal -unspecified.”

F.4 SINGAPORE

Table F.4: Singapore Variant

ITU standards supported by Singapore	Q.701-Q.703, 1992 (White Book) Q.704-Q.707, 1992 (White Book) Q.711-Q.714, 1992 (White Book) Q.761-Q.764, 1992 (White Book)
Singapore specific parameters added	Number Portability (NP)—"Additional Calling Party Number" (0xFA) within the IAM message.

F.5 GERMANY/SWITZERLAND

Table F.5: Germany/Switzerland Variants

ITU standards supported by Germany/Switzerland	Q.701-Q.703, 1992 (White Book) Q.704-Q.707, 1992 (White Book) Q.711-Q.714, 1992 (White Book) Q.761-Q.764, 1992 (White Book)
Germany specific parameter added	Subscriber Priority Class

F.6 ITALY

Table F.6: Italy Variant

ITU standards supported by Italy	Q.701-Q.703, 1992 (White Book) Q.704-Q.707, 1992 (White Book) Q.711-Q.714, 1992 (White Book) Q.761-Q.764, 1992 (White Book)
Italy specific messages added	Call Offering Message (COM) Charge Information (CRG) Identification Request (IDR) Identification Response (IRS)
Unsupported messages	Circuit Group Query (CQM) Circuit Group Query Response (CQR) Confusion (CFN) Facility (FAC) Forward Transfer (FOT) Information (INF) Information Request (INR) Network Resource Management (NRM) Pass Along (PAM) Segmentation (SGM) Unequipped CIC (UCIC) User-to-User Information (USR)
Italy specific parameters added	Charge Band (CBND) Charge Band Request (CBR) Charge Units Indicator (CUI) Crank Back Indicator (CRBI) Incoming Trunk Identity (INTRI)

Table F.6: Italy Variant(Continued)

Unsupported parameters	Automatic Congestion Level (ACL)
	Call Diversion Information (CDUI)
	Call History Information (CHI)
	Call Reference (CRF)
	Connection Request (CONR)
	Echo Control Information (ECI)
	Generic Digits (GND)
	Generic Notification (GNNO)
	Generic Number (GNNU)
	Generic Reference (GNR)
	Information Indicators (INI)
	Information Request Indicators (INRI)
	Location Number (LON)
	Message Compatibility Information (MCOI)
	MLPP Precedence (MLPP)
	Network Specific Facilities (NSF)
	Original Called Number (OCDN)
	Origination ISC Point Code (OISC)
	Parameter Compatibility Information (PCI)
	Propagation Delay Counter (PDC)
	Redirecting Number (RDTG)
	Redirection Number (REDN)
	Redirection Number Restriction (RNR)
	Remote Operations (ROP)
	Service Activation (SAC)
	Transit Network Selection (XNS)
	Transmission Medium Requirement Prime (XMRP)
	Transmission Medium Used (XMU)
	User Service Information Prime (USIP)
	User Teleservice Information (UTI)

F.7 FINLAND

Table F.7: Finland Variant

ITU standards supported by Finland	Q.701-Q.703, 1992 (White Book) Q.704-Q.707, 1992 (White Book) Q.711-Q.714, 1992 (White Book) Q.761-Q.764 and Q.766, 1992 (White Book) SFS 5779 ISDN User Part (ISUP) of the national Signalling System No. 7 for Finland
Finland specific messages added	Charge Acknowledge (CHMA) Charge (CHM) Metering Pulse (MPM)
Unsupported messages	Charge Information (CRG) Continuity (COT) Continuity Check Request (CCR) Forward Transfer (FOT) Identification Request (IDR) Identification Response (IRS) Loopback Acknowledgment (LPA) Pass Along Message (PAM)
Finland specific parameters added	Cancellor SPC (CSPC) CHG Result (CRES) C-number (CNUM) Identity of the Incoming Trunk and Transit Exchange (ITC) Number of Metering Pulses (NOP) One-Time Charge (OTC) Tariff Type (TAR) Time Tariff (TTAR) Volume Tariff (VTAR)

Table F.7: Finland Variant (Continued)

Unsupported parameters	Continuity Indicators (COTI) Free Phone Indicators (FPI) Generic Reference (GNR) Hop Counter (HOP) MCID Request Indicator (MRQ) MCID Response Indicator (MRS)
	MLPP Precedence (MLPP) Origination ISC Point Code (OISC) Service Activation (SAC)

F.8 AUSTRALIA

Table F.8: Australia Variant

ITU standards supported by Australia	Q.701-Q.703, 1992 (White Book) Q.704-Q.707, 1992 (White Book) Q.711-Q.714, 1992 (White Book) Q.761-Q.764, 1992 (White Book) Interconnect ISUP 1.1 (Feb. 21, 1997)
Unsupported messages	Call Modification Complete (CMC) Call Modification Request (CMR) Call Modification Reject (CMRJ) Charge Information (CRG) Circuit Group Query (CQM) Circuit Group Query Response (CQR) Continuity (COT) Continuity Check Request (CCR) Delayed Release (DRS) Facility Accepted (FAA) Facility Request (FAR) Forward Transfer (FOT) Facility Reject (FRJ) Identification Request (IDR) Identification Response (IRS)

Table F.8: Australia Variant (Continued)

Unsupported messages (cont.)	Information Message (INF) Information Request (INR) Loopback Acknowledgment (LPA) Network Resource Management (NLM) Overload Message (OLM) Pass Along Message (PAM) Segmentation Message (SGM) Unidentified Circuit Identification (UCIC) User-to-User Information (USR)
Unsupported parameters	Automatic Delivery Information (ADI) Call Diversion Information (CDUI) Call History Information (CHI) Call Modification Indicators (CMI) Call Reference (CRF) Circuit State Indicator (CTI) Closed User Group Interlock Code (CUGC) Connected Number (CONN) Connection Request (CONR) Continuity Indicators (COTI) Echo Control Information (ECI) Facility Indicators (FCI) Free Phone Indicators (FPI) Generic Digits (GND) Generic Notification (GNNO) Generic Number (GNNU) Generic Reference (GNR) Hop Counter (HOP) Information Indicators (INI) Information Request Indicators (INRI) Location Number (LON)

Table F.8: Australia Variant (Continued)

Unsupported parameters (cont.)	MCID Request Indicator (MRQ) MCID Response Indicator (MRS) MLPP Precedence (MLPP) Network Specific Facilities (NSF) Origination ISC Point Code (OISC) Propagation Delay Counter (PDC) Redirection Number (REDN) Redirection Number Restriction (RNR) Remote Operations (ROP) Service Activation (SAC) Signalling Point Code (SPC) Transit Network Selection (XNS) Transmission Medium Requirement Prime (XMRP) Transmission Medium Used (XMU) User Service Information Prime (USIP) User Teleservice Information (UTI)

F.9 SPAIN

Table F.9: Spain Variant

ITU standards supported by Spain	Q.701-Q.703, 1992 (White Book) Q.704-Q.707, 1992 (White Book) Q.711-Q.714, 1992 (White Book) Q.761-Q.764, 1992 (White Book)
Spain specific messages added	Charging (CHI) Malicious Call (MAL)
Spain specific parameters added	Tariff (TFA) Diversion Information (DIVI) Notification Type (NOT) Virtual Private Network Code (VPNC) Charging (CHI)