



# **Dialogic<sup>®</sup> DSI SS7LD Network Interface Boards**

**Programmer's Manual**

## Copyright and Legal Notice

Copyright © 2010-2013 Dialogic Inc. All Rights Reserved. You may not reproduce this document in whole or in part without permission in writing from Dialogic Inc. at the address provided below.

All contents of this document are furnished for informational use only and are subject to change without notice and do not represent a commitment on the part of Dialogic Inc. and its affiliates or subsidiaries ("Dialogic"). Reasonable effort is made to ensure the accuracy of the information contained in the document. However, Dialogic does not warrant the accuracy of this information and cannot accept responsibility for errors, inaccuracies or omissions that may be contained in this document.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH DIALOGIC® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN A SIGNED AGREEMENT BETWEEN YOU AND DIALOGIC, DIALOGIC ASSUMES NO LIABILITY WHATSOEVER, AND DIALOGIC DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF DIALOGIC PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHT OF A THIRD PARTY.

Dialogic products are not intended for use in certain safety-affecting situations. Please see <http://www.dialogic.com/company/terms-of-use.aspx> for more details.

Due to differing national regulations and approval requirements, certain Dialogic products may be suitable for use only in specific countries, and thus may not function properly in other countries. You are responsible for ensuring that your use of such products occurs only in the countries where such use is suitable. For information on specific products, contact Dialogic Inc. at the address indicated below or on the web at [www.dialogic.com](http://www.dialogic.com).

It is possible that the use or implementation of any one of the concepts, applications, or ideas described in this document, in marketing collateral produced by or on web pages maintained by Dialogic may infringe one or more patents or other intellectual property rights owned by third parties. Dialogic does not provide any intellectual property licenses with the sale of Dialogic products other than a license to use such product in accordance with intellectual property owned or validly licensed by Dialogic and no such licenses are provided except pursuant to a signed agreement with Dialogic. More detailed information about such intellectual property is available from Dialogic's legal department at 1504 McCarthy Boulevard, Milpitas, CA 95035-7405 USA. **Dialogic encourages all users of its products to procure all necessary intellectual property licenses required to implement any concepts or applications and does not condone or encourage any intellectual property infringement and disclaims any responsibility related thereto. These intellectual property licenses may differ from country to country and it is the responsibility of those who develop the concepts or applications to be aware of and comply with different national license requirements.**

Dialogic, Dialogic Pro, Dialogic Blue, Veraz, Brooktrout, Diva, Diva ISDN, Making Innovation Thrive, Video is the New Voice, VisionVideo, Diastar, Cantata, TruFax, SwitchKit, SnowShore, Eicon, Eiconcard, NMS Communications, NMS (stylized), SIPcontrol, Exnet, EXS, Vision, PowerMedia, PacketMedia, BorderNet, inCloud9, I-Gate, ControlSwitch, NaturalAccess, NaturalCallControl, NaturalConference, NaturalFax and Shiva, among others as well as related logos, are either registered trademarks or trademarks of Dialogic Inc. and its affiliates or subsidiaries. Dialogic's trademarks may be used publicly only with permission from Dialogic. Such permission may only be granted by Dialogic's legal department at 1504 McCarthy Boulevard, Milpitas, CA 95035-7405 USA. Any authorized use of Dialogic's trademarks will be subject to full respect of the trademark guidelines published by Dialogic from time to time and any use of Dialogic's trademarks requires proper acknowledgement.

The names of actual companies and products mentioned herein are the trademarks of their respective owners.

This document discusses one or more open source products, systems and/or releases. Dialogic is not responsible for your decision to use open source in connection with Dialogic products (including without limitation those referred to herein), nor is Dialogic responsible for any present or future effects such usage might have, including without limitation effects on your products, your business, or your intellectual property rights.

Publication Date: January 2013

Document Number: U01OPL

## Revision History

Issue	Date	Description
3	10-Jan-13	Addition of drop and insert functionality with MVD_MSG_RESETSWX and MVD_MSG_SC_CONNECT messages. General updates including monitoring modes, T1/E1 configuration (on a per board basis), return status values and addition of 8 link license.
2	13-Jun-11	GA release.
1	17-Dec-10	Initial release.

Note: The current version of this guide can be found at:  
<http://www.dialogic.com/support/helpweb/signaling>

## Contents

<b>Revision History</b> .....	<b>3</b>
<b>1 Introduction</b> .....	<b>5</b>
1.1 Related Information .....	5
<b>2 Specification</b> .....	<b>6</b>
2.1 Product Identifier.....	6
2.2 Capacity .....	6
2.3 Host Interface .....	7
2.4 T1/E1 Digital Trunk Interfaces .....	7
2.5 CT bus Interface.....	7
2.6 Protocol Resource Support .....	7
2.7 Visual Indicators.....	8
2.8 Power Requirements .....	9
2.9 Physical Specification .....	9
2.10 Environmental Specification .....	9
2.11 Safety, EMC and Telecommunications Specifications .....	10
2.12 Reliability .....	10
2.13 Software Licenses.....	10
2.13.1 Run Modes.....	11
2.14 Regulatory and Geographic Considerations .....	11
<b>3 Configuration and Operation</b> .....	<b>13</b>
3.1 Board Code File .....	13
3.2 Running MTP3 and ISUP .....	13
3.2.1 Running MTP3 .....	14
3.2.2 Running ISUP.....	14
3.3 Monitoring .....	14
3.4 Using the Dialogic® SyncRoute Connector .....	15
3.5 Switching Timeslots between LIUs .....	15
3.5.1 Cross Connect Operation using Messages .....	16
3.5.2 Interconnecting LIUs using STREAM_XCON .....	16
<b>4 Message Reference</b> .....	<b>18</b>
4.1 Message Summary Table .....	19
4.2 Software Module IDs for SS7LD Board.....	21
4.3 Message Status Summary.....	22

4.4	General Configuration Messages .....	23
4.4.1	SSD_MSG_RESET - SSD Reset Request .....	24
4.4.2	SSD_MSG_RST_BOARD - Board Reset Request .....	25
4.4.3	SSD_MSG_BOARD_INFO - Board Information Request .....	26
4.4.4	MGT_MSG_CONFIG0 - Board Configuration Request.....	28
4.4.5	MGT_MSG_L1_CONFIG - Layer 1 Configuration Request.....	30
4.4.6	MGT_MSG_L1_END - Layer 1 Configuration End .....	32
4.5	Hardware Control Messages .....	33
4.5.1	LIU_MSG_CONFIG - LIU Configuration Request .....	34
4.5.2	LIU_MSG_CONTROL - LIU Control Request .....	37
4.5.3	LIU_MSG_R_CONFIG - LIU Read Configuration Request.....	38
4.5.4	LIU_MSG_R_CONTROL - LIU Read Control Request.....	39
4.5.5	MVD_MSG_CNFCLOCK - Configure Clock Request .....	41
4.5.6	MVD_MSG_CLOCK_PRI - Configure Clock Priority Request .....	44
4.5.7	MVD_MSG_RESETSWX - Reset Switch Request.....	45
4.5.8	MVD_MSG_SC_CONNECT - Connect Request .....	46
4.6	Signaling Interface Messages .....	49
4.7	Event Indication Messages.....	50
4.7.1	SSD_MSG_STATE_IND - Board Status Indication.....	51
4.7.2	MVD_MSG_CLK_IND - Clock Event Indication.....	52
4.7.3	MVD_MSG_LIU_STATUS - LIU Status Indication .....	53
4.7.4	DVR_MSG_TRACE_MASK - Set Trace Mask Request .....	54
4.8	Status Request Messages .....	57
4.8.1	LIU_MSG_R_STATE - LIU Read State Request .....	58
4.8.2	LIU_MSG_R_STATS - LIU Read Statistics Request.....	59
4.8.3	MVD_MSG_R_CLK_STATUS - Clock Status Request .....	61
4.8.4	MGT_MSG_R_BRDINFO - Read Board Info Request.....	62
4.8.5	DVR_MSG_R_L1_STATS - Link Statistics Request .....	64
<b>Appendix A. Protocol Configuration Using Discrete Messages .....</b>		<b>66</b>
A.1	Protocol Configuration Using Individual Messages .....	66
A.2	Monitoring Configuration Using Individual Messages.....	69

## Tables

Table 1.	Dialogic® DSI SS7LD Software Licenses .....	11
Table 2.	Link License Resource Requirements.....	11
Table 3.	Message Summary .....	19
Table 4.	DSI SS7LD Board Software Module IDs .....	21
Table 5.	Message Status Responses .....	22

## Figures

Figure 1.	End Bracket Showing Connectors and User LEDs .....	8
Figure 2.	Physical Location of SyncRoute Connector .....	15
Figure 3.	Drop and Insert .....	17
Figure 4.	Protocol Configuration Message Sequence Diagram .....	68

# 1 Introduction

Dialogic® DSI SS7LD Network Interface Boards are specialized T1/E1 SS7 signaling boards suitable for use in PCI Express form factor systems. The boards use the common Dialogic® DSI software API to the application that enables applications to be ported easily between hardware architectures.

The boards provide a hardware platform to enable running Dialogic® DSI Protocol Stacks for the realization of Signaling System Number 7 signaling nodes. In addition, the DSI SS7LD Boards can be used to build low cost monitoring applications. The boards can be used under the Linux, Solaris SPARC and Solaris x86 operating systems.

This manual is the Programmer's Manual for the Dialogic® DSI SS7LD range of network interface boards. It is targeted for system developers who are integrating the boards and who have chosen to develop applications that use the underlying DSI Protocol Stack. The manual includes information on the operation of the boards and the SS7 software stack.

The manual should be used in conjunction with the *Dialogic® Distributed Signaling Interface Components - Software Environment Programmer's Manual* and the appropriate Installation Guide and Regulatory Notice for the board. These and other supporting documentation, including the Programmer's Manuals for the individual protocol modules, are listed in Section 1.1, Related Information.

Note: Users of the Dialogic® DSI SS7MD, DSI SPCI4, and DSI SPCI2S Network Interface Boards should refer to separate documentation that covers those boards.

## 1.1 Related Information

Refer to the following for related information:

- *Dialogic® Distributed Signaling Interface Components - Software Environment Programmer's Manual*
- *Dialogic® DSI SS7LD Network Interface Board Installation Guide*
- *Dialogic® DSI SS7LDH4Q Network Interface Board Regulatory Notices*
- *Dialogic® SS7 Protocols MTP2 Programmer's Manual*
- *Dialogic® SS7 Protocols MTP3 Programmer's Manual*
- *Dialogic® SS7 Protocols ISUP Programmer's Manual*
- *Dialogic® DSI Protocol Stacks SNMP User Manual*
- *Dialogic® DSI Protocol Stacks - Host Licensing User Guide*

Current software and documentation supporting Dialogic® DSI SS7LD Boards as well as product data sheets and information on Dialogic® DSI SS7 product and services is available at <http://www.dialogic.com/support/helpweb/signaling>.

## 2 Specification

This section provides information about the physical specification and capacity of the product:

- Product Identifier
- Capacity
- Host Interface
- T1/E1 Digital Trunk Interfaces
- CT bus Interface
- Protocol Resource Support
- Visual Indicators
- Power Requirements
- Physical Specification
- Environmental Specification
- Safety, EMC and Telecommunications Specifications
- Reliability
- Software Licenses
- Regulatory and Geographic Considerations

### 2.1 Product Identifier

The Dialogic® DSI SS7LD Network Interface Board product family includes a single PCI Express form factor board with the following designation:

Dialogic® DSI SS7LD Network Interface Board - SS7LDH4Q

Within this document, the generic term "SS7LD" is used to cover the SS7LDH4Q product.

### 2.2 Capacity

The capacity of the DSI SS7LDH4Q board is described as follows:

- **Digital interfaces**

— Four T1 or E1 (software selectable per board)

- **SS7 links**

Able to terminate up to 4 Q.703 Low Speed Links (LSL) at 64kbit/s, 56kbit/s or 48kbit/s or monitor up to 8 simplex Q.703 LSL (eg. up to 4 links duplex).

- **Dialogic® DSI Protocol Stacks**

MTP2 on board; other protocols are host-based.

MTP3 and ISUP optionally run as host-based protocols embedded within the ssdl binary.

## 2.3 Host Interface

The DSI SS7LDH4Q board has a x1 PCI Express connector. It can be installed in x1, x4, x8, or x16 PCI Express slots.

## 2.4 T1/E1 Digital Trunk Interfaces

The properties of the four T1/E1 digital trunk interfaces are as follows:

- **Standard**

- Four interfaces software configurable as either T1 or E1 on a per board basis.
- High impedance software selectable on a per port basis

- **Pulse mask**

- T1: ANSI T1.403
- E1: ITU-T G.703

- **Data rate**

- T1: 1544 kbits/s  $\pm$  50 ppm
- E1: 2048 kbits/s  $\pm$  50 ppm

- **Frame format**

- T1: D3/D4 and ESF
- E1: E1 and E1-CRC4

- **Line codes**

- T1: B8ZS and AMI
- E1: HDB3 and AMI

- **Connector type**

- RJ-48C

## 2.5 CT bus Interface

The board supports the Dialogic® SyncRoute Connector which is a 16-pin SyncRoute Connector providing Limited H.100 computer telephony capabilities allowing a common telecom clock to be shared across multiple boards. The SyncRoute bus supports the clock/frame sync synchronization and clock fall back features of the CT bus.

## 2.6 Protocol Resource Support

When used in a signaling node, the DSI SS7LDH4Q board supports the Message Transfer Part (MTP) running on the board and optionally other protocols including MTP3, ISUP, TUP, SCCP, TCAP, MAP, INAP and IS41 running on the host. The protocols are enabled by software licenses. See Section 2.13, "Software Licenses" on page 10.

The DSI SS7LDH4Q board supports passive monitoring of HDLC format data links including, for example, SS7, LAPB, LAPD, ISDN, and DPNSS. In this mode, the received messages are directly reported to the application. For more information on link monitoring, see Section 3.3 "Monitoring" on page 14.

It is possible to use monitor and receive-transmit protocol operations concurrently on the same signaling board.

## 2.7 Visual Indicators

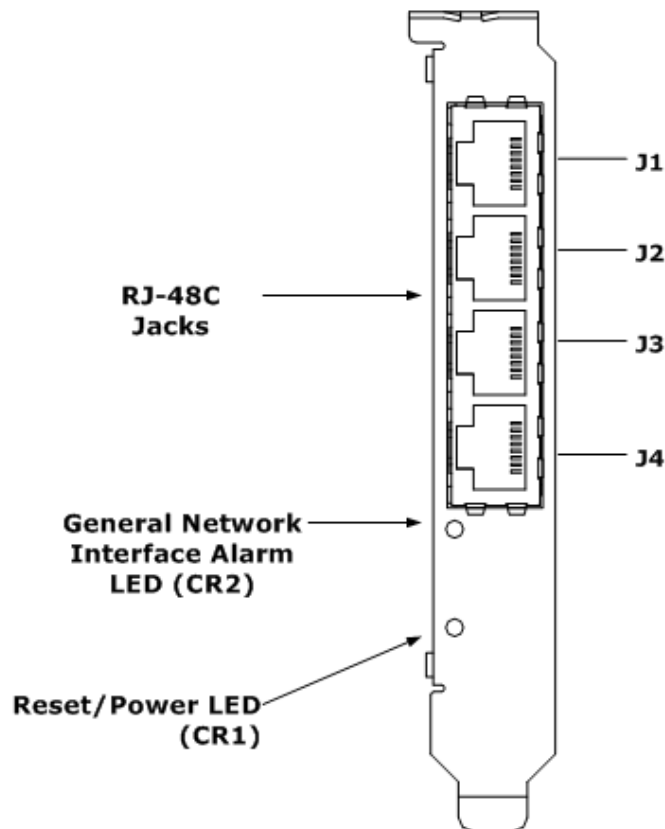
The DSI SS7LDH4Q board includes the following visual indicators:

**General Network Interface Alarm LED:** When lit (yellow), indicates that an alarm condition is present on one or more of the trunks. When unlit, alarm conditions are cleared.

**Reset/Power LED:** When lit (green), indicates that board power is good. When unlit, either power has not been applied to the board, or the board has detected that one or more of the on-board-generated voltages are not correct.

When lit (red), it indicates that the board is in a reset state due to either PCIe Reset from the edge connector being active or one or more of the on-board-generated voltages are not correct.

**Figure 1. End Bracket Showing Connectors and User LEDs**





## 2.8 Power Requirements

Power requirements are described as follows:

- +12 VDC power  
0.25 A max.
- +3.3 VDC power  
1.8 A max.
- Power dissipation  
8.94 W max.

## 2.9 Physical Specification

- **Form factor**

x1 lane, standard height, half length PCIe board

- **Dimensions**

**Board**

Length	167.65 mm (6.60 inches)
Height	111.15 mm (4.376 inches)

**Packaged**

Length	270 mm (10.7 inches)
Width	225 mm (8.9 inches)
Height	40 mm (1.6 inches)

- **Weight**

Board	122 grams
Packaged board	340 grams

## 2.10 Environmental Specification

Environmental specification is described as follows:

- Operating temperature range  
+0°C to +55°C
- Storage temperature range  
-20°C to +70°C
- Humidity

Non-operating 50% to 85% relative humidity, for 25° C – 70° C.

- Altitude  
0 to 15,000 ft
- Vibration  
0.1 g, 5 to 100 Hz

- Shock

Packaged equipment drop test 36 in (914 mm)

## 2.11 Safety, EMC and Telecommunications Specifications

Safety, EMC and telecommunications specification information is provided by the following:

- Dialogic® DSI SS7LDH4Q Network Interface Board Regulatory Notices

Supplied with each product and provides a list of the specifications to which the DSI SS7LDH4Q board conforms.

- International Declaration of Conformity

See <http://www.dialogic.com/declarations>.

- Country-Specific Approvals

See the Global Product Approvals list at <http://www.dialogic.com/declarations>.

Alternatively, contact your Dialogic technical sales representative for more information.

## 2.12 Reliability

Product reliability is described by:

- MTBF Predication

190,000 hours per Telcordia Issue I, Method I, Case I

- Warranty

See Dialogic® Telecom Products Warranty Information at <http://www.dialogic.com/warranties>.

## 2.13 Software Licenses

The DSI SS7LD code file supports different MTP2 link densities on the board. These are enabled using a Host Software License that is to be ordered at the same time as the hardware. The Host Software License licenses a specific number of link resources on the host that may be shared between boards in the same chassis.

For details on how to activate the host license, please refer to Dialogic® DSI Protocol Stacks - Host Licensing User Guide U32SSS see section 1.1 Related Information.

A combination of link types (provided they are supported by the board's run mode) may be configured by the host (on any board) provided the required link resources are available. A configured link's resources are freed when either the link is unconfigured or the board on which the link is currently active is reset.

The following table shows the available licenses:

**Table 1. Dialogic® DSI SS7LD Software Licenses**

Software License	Code	Capacity			
		Maximum Number of Link Sets	Maximum Number of Routes	Maximum Number of Circuit Groups	Maximum Numbers of Circuits
SS7SBLDISUP1K	ISUP (Small)	2	64	44	1024
SS7SBLDISUP2K	ISUP (Regular)	4	64	64	2048
SS7SBLDISUP4K	ISUP (Large)	4	64	128	4096
SS7SBLDISUP4K8L	ISUP	8	64	128	4096

The number of link resources required for each link type is shown below:

**Table 2. Link License Resource Requirements**

Link Type	Resources Required
LSL (64Kb / 56Kb / 48Kb)	1
Monitored LSL	0.5

### 2.13.1

#### Run Modes

The run\_mode parameter in either the SS7\_BOARD command or the Board Reset Request message is used to set the operating mode for the ss7l binary and must be the same for all boards within the system.

In the event that the user wishes to change the run mode, it is necessary to restart gctload so that ssdl is restarted.

The following table shows the protocol that will run within ssdl for each run mode. If not running within ssdl then the user can run the protocol as a stand-alone host binary. All protocols that run embedded within ssdl use their own message queue so they require a LOCAL entry in the system.txt file.

Run Mode	Protocols running embedded within ssdl	Optional Host Protocols
MTP2	None	MTP3, ISUP, SCCP etc
MTP	MTP3	ISUP, SCCP etc
ISUP	MTP3 and ISUP	SCCP etc

### 2.14

#### Regulatory and Geographic Considerations

Certain functions of the Dialogic® DSI SS7LD Network Interface Board, although implemented in hardware, have selectable options that are configured by the software. A user or integrator must consider the requirements of the application when choosing these settings, but must also consider any local regulatory requirements for the intended deployment location to provide a compliant overall system. As an aid to this process, the table below details some of the areas where the correct selection of configuration options may be required.

Configuration Area		Configuration Options
T1/E1 Ports	Interface type	liu_type parameter in LIU_CONFIG command
	Pulse shape	liu_type parameter in LIU_CONFIG command
	Line code	line_code parameter in LIU_CONFIG command
	Frame format	frame_format parameter in LIU_CONFIG command
	CRC/E-bit operation	CRC_mode parameter in LIU_CONFIG command
	Clock priorities	flags parameter in SS7_BOARD command and options parameter in LIU_CONFIG command
CT Bus	Master/Slave configuration	flags parameter in SS7_BOARD command
	Bus termination	flags parameter in SS7_BOARD command
Links	Link termination or monitoring mode	MTP_LINK or MONITOR_LINK commands

Note: For details on these configuration commands please refer to *Dialogic® Distributed Signaling Interface Components - Software Environment Programmer's Manual*

## 3 Configuration and Operation

Before attempting software configuration, you should gain an understanding of the flexibility of the protocol stack, the run-time options that exist and the mechanisms that are used to select specific features. These are explained in the *Dialogic® Distributed Signaling Interface Components - Software Environment Programmer's Manual* which also describes the basic principles of modules and message passing.

This section provides additional detail on the board-specific product options which are available.

This section provides information about:

- Board Code File
- Running MTP3 and ISUP
- Monitoring
- Using the Dialogic® SyncRoute Connector
- Switching Timeslots between LIUs

### 3.1 Board Code File

The firmware for the SS7LD board is contained within a code file called `ss7.dc7` and must be downloaded to the board at run time. The host-based `ssdl` utility manages this download operation.

The code file is distributed as part of the Dialogic® DSI Development Package and can be downloaded from the Dialogic website.

The use of the code file requires a host license which enables the software to run on the board, details on how to use a Host License are given in the *Dialogic® Distributed Signaling Interface Components Host Licensing User Guide*.

### 3.2 Running MTP3 and ISUP

The Dialogic® DSI MTP2 Layer protocol module runs on the SS7LD board. The Dialogic® DSI MTP3 and the Dialogic® DSI ISUP Layer protocol modules may be run either as individual host protocols, which need to be separately licensed, or as embedded protocols within the `ssdl` binary using a 'bundled license'. The use of the embedded MTP3 and ISUP binaries provided a lower cost solution for low density deployments.

Other SS7 protocol modules (eg. TUP, SCCP, TCAP, MAP, INAP and IS41) are not embedded within the `ssdl` binary and need to be run separately.

When running MTP3 or ISUP embedded within `ssdl`, the `run_mode` should be set to MTP3 or ISUP and it is NOT necessary (or appropriate) to have a `FORK_PROCESS` command for the MTP3 or ISUP host binaries.

When running MTP3 and ISUP as separate host binaries `run_mode` must be set to MTP2 and a `FORK_PROCESS` command must be used for MTP3 and ISUP host binaries.

### 3.2.1 Running MTP3

Whether MTP3 is running as an individual host binary or embedded within ssdl it is always necessary to declare MTP3 as a LOCAL module in the system.txt configuration file as follows:

```
LOCAL 0x22      * MTP3 module
```

It is not necessary (or appropriate) to use the REDIRECT command for MTP3 when using the SS7LD board.

### 3.2.2 Running ISUP

When using the ISUP protocol, whether ISUP is running as an individual host binary or embedded within ssdl it is always necessary to declare ISUP as a LOCAL module in the system.txt configuration file as follows:

```
LOCAL 0x23      * ISUP module
```

It is not necessary (or appropriate) to use the REDIRECT command for ISUP when using the SS7LD board.

Note: For further information on system.txt files refer to the *Dialogic® Distributed Signaling Interface Components - Software Environment Programmer's Manual*.

## 3.3 Monitoring

The board can be used in receive only mode for the purposes of passive monitoring of the signaling traffic.

In this mode the board is able to monitor many HDLC based signaling protocols including SS7, LAPB, Q.921 (ISDN PRI) and DPNSS. The protocol should have a minimum frame length (excluding flags) of 5 octets and a maximum of 278 octets, and must use the CRC polynomial (x16 + x12 + x5 + 1). When operating in monitoring mode, the 3rd and successive identical frames may be filtered.

It is possible to monitor and terminate SS7 links on the same signaling board.

For receive only operation, the board allows the T1/E1 interfaces to be configured in any of the following modes:

- Normal terminated impedance
- High impedance
- Protected Monitoring Point mode (preferred mode of operation for monitoring).

The use of a Protective Monitoring Point provides increased protection to the signal being monitored. When using High Impedance mode care should be taken to avoid long cable runs as this can result in poor signal quality due to signal reflections.

Configuration details are described in section 4.5.1 LIU\_MSG\_CONFIG - LIU Configuration Request on page 34).

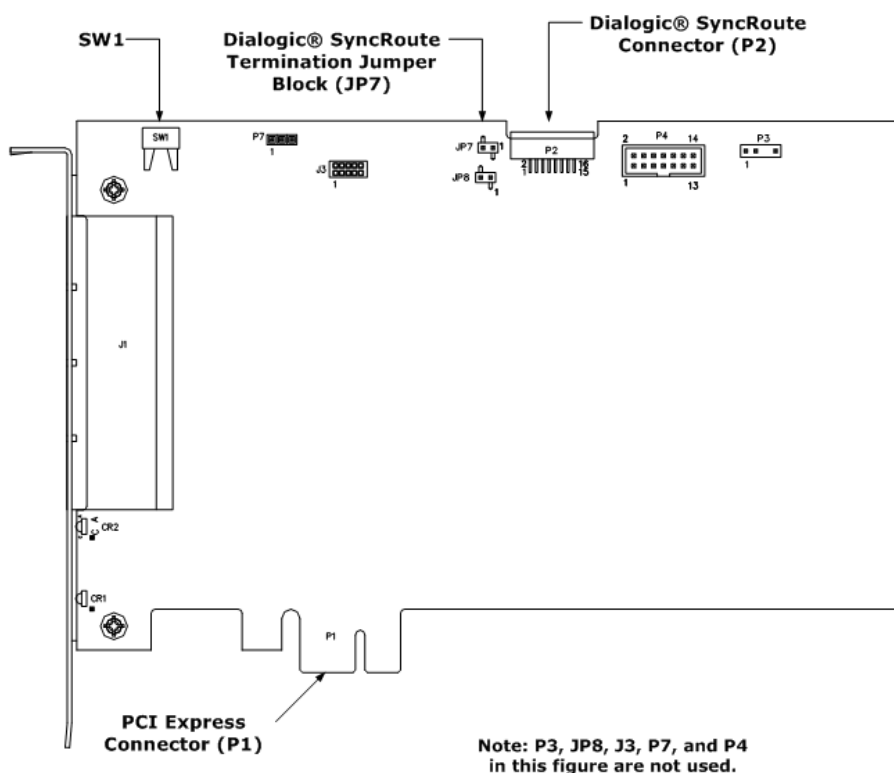
### 3.4 Using the Dialogic® SyncRoute Connector

The Dialogic® DSI SS7LDH4Q Network Interface Boards have a SyncRoute connector which permits multiple boards within a single system to be synchronized from a single clock source. The clock source is selected using the MVD\_MSG\_CNFCLOCK message and clock priority is set using the MVD\_MSG\_CLOCK\_PRI message.

If you are connecting multiple boards via the SyncRoute bus connector, the bus signal should be terminated on the boards that are located at the ends of the cable. All other boards should be left in their factory default configuration with the SyncRoute termination pins not linked.

To terminate the SyncRoute, install a link clip over the pair of JP7 pins.

**Figure 2. Physical Location of SyncRoute Connector**



### 3.5 Switching Timeslots between LIUs

The Dialogic® DSI SS7LD Boards support multiple T1/E1 Line Interface Units (LIUs). The onboard signaling processor handles the SS7 signaling timeslots, while the remaining circuits (voice or data bearer circuits) can optionally be switched to another onboard LIU for distribution to other boards.

Communication between the application and the board is message-based. Initial configuration is typically handled by the s7\_mgt protocol configuration utility that takes commands from the config.txt protocol configuration file and generates the necessary configuration messages for the board. Subsequent operation is entirely message driven, with messages being passed in both directions between the board and the application.

One of the roles of the application is to control the dynamic switching between LIUs. This section provides details of how to interface with the cross connect switch, including the initial (static) configuration and the subsequent (dynamic) switching. The operation of the switching interface is described using:

- MVD\_MSG\_SC\_CONNECT messages.
- STREAM\_XCON config.txt commands.

Note: The use of the STREAM\_XCON command and others is explained in the DSI Software Environment Programmer's Manual.

### **3.5.1 Cross Connect Operation using Messages**

The application controls initialisation and dynamic changes to switching by sending the MVD\_MSG\_SC\_CONNECT message to the board. The message is used to make a connection between a specified timeslot in a source stream and a specified timeslot in the local stream.

The message may be used to set:

- Fixed pattern on local timeslot.
- Simplex local timeslot to local timeslot.
- Duplex local timeslot to local timeslot.
- Remove duplex local timeslot to local timeslot.

Multiple messages should be used to cross connect between LIUs, one per timeslot.

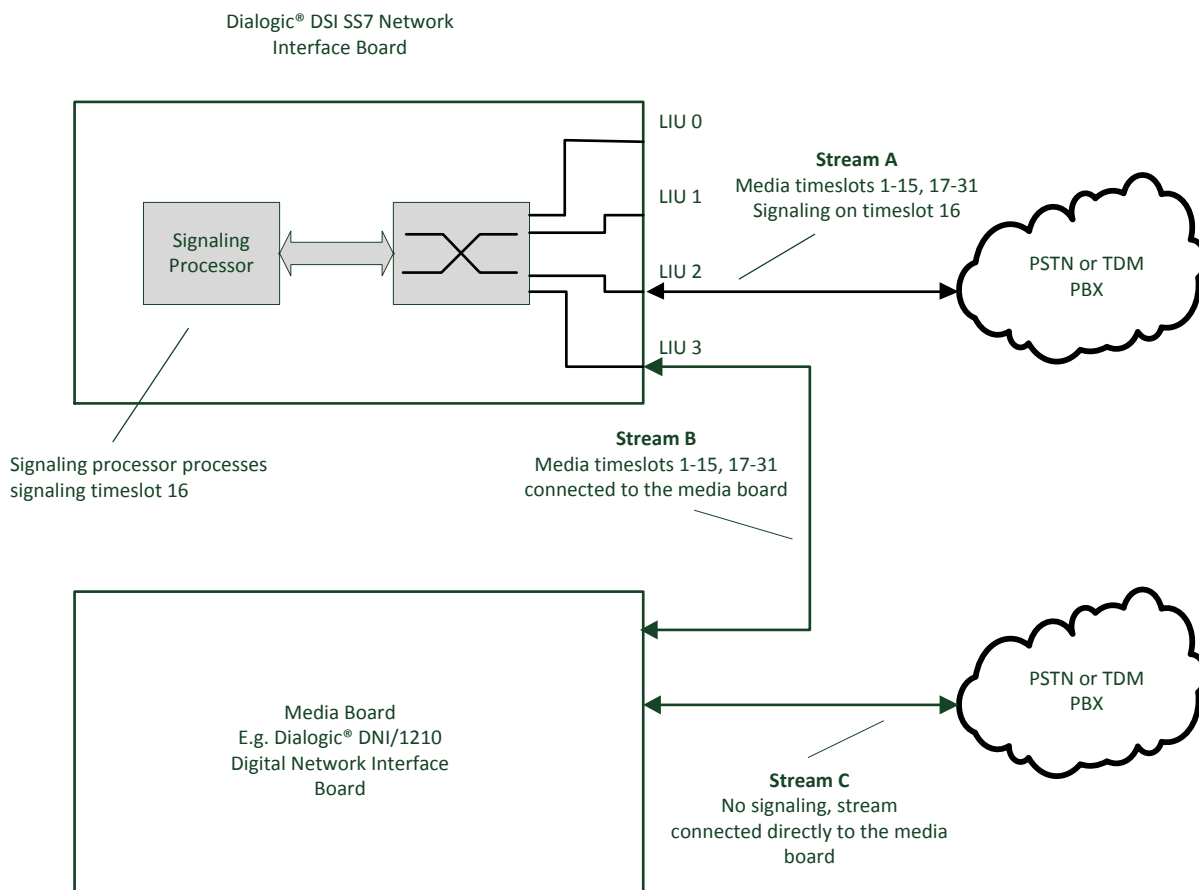
Connections may be removed using the MVD\_MSG\_RESETSWX message. Use of this message will reset all connections previously established.

### **3.5.2 Interconnecting LIUs using STREAM\_XCON**

Interconnection of two Line Interface Units (LIUs) on the board is also supported through the STREAM\_XCON command which controls the cross connect switch on the signaling board, enabling the cross connection of timeslots between any two LIUs within the board. This command simplifies the cross connection enabling a group of timeslots on one LIU to be directly mapped to the same numbered timeslots on a second LIU on the same board using a single command. A typical usage of the STREAM\_XCON command is shown in Figure 3 which implements Drop and Insert functionality.



**Figure 3. Drop and Insert**



Use a single `STREAM_XCON` command to cross-connect the input and output timeslot. The timeslot mask selects which timeslots are cross-connected (0xffffffe).

Example `s7_mgt` configuration command:

```
STREAM_XCON 0 2 3 3 0xffffffe 0
```

## 4 Message Reference

This section describes the individual messages that may be sent to or received from a Dialogic® DSI SS7LD Network Interface Board. Some messages are sent by the user's application software, while others are sent by utility programs such as the s7\_mgt protocol configuration utility.

Prior to sending any message to the board, the application should call the **GCT\_set\_instance( )** library function to select which board the message will be sent to. After receiving a message from the board, the application should call the **GCT\_get\_instance( )** library function to determine which board the message came from. These library functions are described in the *Dialogic® Distributed Signaling Interface Components - Software Environment Programmer's Manual*.

The various messages used are grouped in the following categories:

- General Configuration Messages
- Hardware Control Messages
- Signaling Interface Messages
- Event Indication Messages
- Status Request Messages

The following section provides a summary of all messages used in this manual. The format of the message header for all messages is detailed in the *Dialogic® Distributed Signaling Interface Components - Software Environment Programmer's Manual*.

## 4.1 Message Summary Table

The following table lists, by message type, the messages described in this manual.

**Table 3. Message Summary**

Message Type	Mnemonic	Description
0x0201	MGT_MSG_SS7_STATE*	Link State Indication*
0x0202	MGT_MSG_SS7_EVENT*	MTP2 Q.791 Event Indication*
0x0301	MGT_MSG_MTP_EVENT*	MTP3 Q.752 Event Indication*
0x06a0	SSD_MSG_STATE_IND	Board Status Indication
0x0e01	MVD_MSG_LIU_STATUS	LIU Status Indication
0x0e23	MVD_MSG_CLK_IND	Clock Event Indication
0x1213		Confirmation of SS7_MSG_TRACE_MASK*
0x1e36		Confirmation of LIU_MSG_R_STATS
0x1e37		Confirmation of LIU_MSG_R_CONFIG
0x1e38		Confirmation of LIU_MSG_R_CONTROL
0x1e39		Confirmation of LIU_MSG_R_STATE
0x2214		Confirmation of SS7_MSG_R_STATS*
0x2215		Confirmation of SS7_MSG_R_STATE*
0x2236		Confirmation of DVR_MSG_R_L1_STATS
0x2e04		Confirmation of MVD_MSG_R_CLK_STATUS
0x2f0d		Confirmation of MGT_MSG_R_BRDINFO
0x3200		Confirmation of SS7_MSG_RESET*
0x3203		Confirmation of SS7_MSG_CONFIG*
0x331b		Confirmation of DVR_MSG_TRACE_MASK
0x3680		Confirmation of SSD_MSG_RESET
0x3681		Confirmation of SSD_MSG_RST_BOARD
0x3689		Confirmation of SSD_MSG_BOARD_INFO
0x3e20		Confirmation of MVD_MSG_CNFCLOCK
0x3e21		Confirmation of MVD_MSG_CLOCK_PRI
0x3e34		Confirmation of LIU_MSG_CONFIG

Message Type	Mnemonic	Description
0x3e35		Confirmation of LIU_MSG_CONTROL
0x3f10		Confirmation of MGT_MSG_CONFIG0
0x3f17		Confirmation of MGT_MSG_L1_CONFIG
0x3f18		Confirmation of MGT_MSG_L1_END
0x5213	SS7_MSG_TRACE_MASK*	Set Trace Mask Request*
0x5e36	LIU_MSG_R_STATS	LIU Read Statistics Request
0x5e37	LIU_MSG_R_CONFIG	LIU Read Configuration Request
0x5e38	LIU_MSG_R_CONTROL	LIU_MSG_R_CONTROL
0x5e39	LIU_MSG_R_STATE	LIU Read State Request
0x6136	DVR_MSG_R_L1_STATS	Link Statistics Request
0x6214	SS7_MSG_R_STATS*	MTP2 Link Statistics Request*
0x6215	SS7_MSG_R_STATE*	MTP2 Read State Request*
0x631e	MTP_MSG_R_RT_STATUS*	Route Status Request*
0x6e04	MVD_MSG_R_CLK_STATUS	Clock Status Request
0x6f0d	MGT_MSG_R_BRDINFO	Read Board Info Request
0x7200	SS7_MSG_RESET*	MTP2 Module Reset Request*
0x7203	SS7_MSG_CONFIG*	MTP2 Link Configuration Request*
0x731b	DVR_MSG_TRACE_MASK	Set Trace Mask Request
0x7680	SSD_MSG_RESET	SSD Reset Request
0x7681	SSD_MSG_RST_BOARD	Board Reset Request
0x7689	SSD_MSG_BOARD_INFO	Board Information Request
0x7e00	MVD_MSG_RESETSWX	Reset Switch Request
0x7e1f	MVD_MSG_SC_CONNECT	Connect Request
0x7e20	MVD_MSG_CNFCLOCK	Configure Clock Request
0x7e21	MVD_MSG_CLOCK_PRI	Configure Clock Priority Request
0x7e34	LIU_MSG_CONFIG	LIU Configuration Request
0x7e35	LIU_MSG_CONTROL	LIU Control Request
0x7f10	MGT_MSG_CONFIG0	Board Configuration Request

Message Type	Mnemonic	Description
0x7f17	MGT_MSG_L1_CONFIG	Layer 1 Configuration Request
0x7f18	MGT_MSG_L1_END	Layer 1 Configuration End
0x830a		Confirmation of MTP_MSG_ACT_SL *
0x830b		Confirmation of MTP_MSG_DEACT_SL *
0x8403	MTP_MSG_PAUSE*	MTP Pause Indication*
0x8404	MTP_MSG_RESUME*	MTP Resume Indication*
0x8405	MTP_MSG_STATUS*	MTP Status Indication*
0x8f01	API_MSG_RX_IND*	Received Message Indication*
0xc30a	MTP_MSG_ACT_SL*	MTP Link Activation Request*
0xc30b	MTP_MSG_DEACT_SL*	MTP Link Deactivation Request*
0xcf00	API_MSG_TX_REQ *	Message for Transmission Request*

\* For more information about:

SS7\_MSG\_XXX messages, consult the *Dialogic® SS7 Protocols MTP2 Programmer's Manual*.

MTP\_MSG\_XXX messages, consult the *Dialogic® SS7 Protocols MTP3 Programmer's Manual*.

## 4.2 Software Module IDs for SS7LD Board

The following table details the software modules IDs for all modules running on the SS7LD board:

**Table 4. DSI SS7LD Board Software Module IDs**

Mnemonic	Value	Description
MGMT_TASK_ID	0x8e	SS7LD Board Management Module
MVD_TASK_ID	0x10	SS7LD LIU and Switch Management Module
SS7_TASK_ID	0x71	MTP2 Module
DVR_ALT_TASK_ID	0x61	Signaling Driver Module

## 4.3 Message Status Summary

The following table shows the valid responses when a response request (rsp\_req) is requested in a message.

**Table 5. Message Status Responses**

Mnemonic	Value	Description
SDE_MSG_OK	0x00	Success
SDE_BAD_ID	0x01	Inappropriate or invalid id in request message
SDE_BAD_STATE	0x02	Message received in wrong state
SDE_BAD_SIG	0x03	Bad signal received
SDE_UNEX_SIG	0x04	Unexpected signal received
SDE_BAD_MSG	0x05	Unsupported message received
SDE_BAD_PARAM	0x06	Invalid parameters contained in message
SDE_NO_RESOURCES	0x07	Insufficient internal message resources
SDE_INVALID_NC	0x08	Invalid Network Context
SDE_INVALID_VERSION	0x09	Message version is invalid
SDE_LICENCE_ERR	0x0e	Failure due to a licensing restriction

## 4.4 General Configuration Messages

General configuration messages are typically issued by the s7\_mgt protocol configuration utility, in which case they need not, and should not, be generated by the user application software.

If the user elects not to use the s7\_mgt protocol configuration utility, it is necessary for the application to build and send messages that:

- configure the ssdl module
- reset each board
- configure each board
- optionally configure additional routes

The messages in the general configuration category include:

- SSD\_MSG\_RESET - SSD Reset Request
- SSD\_MSG\_RST\_BOARD - Board Reset Request
- SSD\_MSG\_BOARD\_INFO - Board Information Request
- MGT\_MSG\_CONFIG0 - Board Configuration Request
- MGT\_MSG\_L1\_END - Layer 1 Configuration End

### 4.4.1 SSD\_MSG\_RESET - SSD Reset Request

#### Synopsis

Sets up ssd module run-time options at initialization time.

**Note:** When using the s7\_mgt protocol configuration utility, this message is generated by s7\_mgt and should not be generated by the user.

#### Format

MESSAGE HEADER		
Field Name		Meaning
type		SSD_MSG_RESET (0x7680)
id		0
src		Sending module ID
dst		SSD_module_ID
rsp_req		Used to request a confirmation.
hclass		0
status		0
err_info		0
len		24
PARAMETER AREA		
Offset	Size	Name
0	3	Reserved. Set to 0.
3	1	mgmt_id
4	18	Reserved. Set to 0.
22	2	num_boards

#### Description

This message is used during initialization by the application to reset the ssd module and set up its run-time parameters.

The confirmation message (if requested) indicates success with a **status** value of 0.

#### Parameters

##### mgmt\_id

The module ID of the management module to which SSD should send board status indications.

##### num\_boards

The maximum number of boards that ssdl is required to manage. This should not exceed 16.



## 4.4.2 SSD\_MSG\_RST\_BOARD - Board Reset Request

### Synopsis

Reset a single board and download a code file.

**Note:** When using the s7\_mgt protocol configuration utility, this message is generated by s7\_mgt and should not be generated by the user.

### Format

MESSAGE HEADER		
Field Name	Meaning	
type	SSD_MSG_RST_BOARD (0x7681)	
id	board_id	
src	Sending module ID	
dst	SSD_module_ID	
rsp_req	Used to request a confirmation.	
hclass	0	
status	0	
err_info	0	
len	26 or 78	
PARAMETER AREA		
Offset	Size	Name
0	2	board_type
2	4	Reserved. Must be set to 0.
6	18	code_file
24	2	run_mode
26	2	options
28	50	code_file_ext

### Description

This message is used by the application during initialization (or reconfiguration) to reset a board and download the code file that contains the operating software for the board. The download operation is supervised by the device driver that reads the binary format code file and transfers it to the board.

The confirmation message (if requested) indicates success with a **status** value of 0. This implies that the reset operation has commenced, but does not imply completion. The application should then wait until a Board Status Indication message is received that indicates either successful completion of the reset and download operation or failure during the procedure.

### Parameters

#### board\_type

The type of board to be reset. This must be set to 32 for SS7LD Boards.

#### code\_file

Null terminated string giving the filename of the code file to be downloaded to the board.

**run\_mode**

Run mode selects the protocols to be run within ssdl (subject to appropriate run-time licenses in place) and should be set to the value shown in the following table:

Run Mode	Value	Protocols running embedded within ssdl	Optional Host Protocols
MTP2	2	None	MTP3, ISUP, SCCP etc
MTP	3	MTP3	ISUP, SCCP etc
ISUP	4	MTP3 and ISUP	SCCP etc

**options**

Bit 0 – Enable SNMP for this board.

Bit 3 – Setting this option forces timestamps to be monotonically increasing.

**code\_file\_ext**

This parameter is an alternative way of specifying the cofe file filename which allows longer filenames to be used. It should contain a null terminated filename (optionally including the path). If the code\_file parameter is set to a null value, the code\_file\_ext parameter will be used. If the code\_file parameter is set to a value other than null, this will take precedence and the data in the code\_file\_ext parameter will be discarded.

**4.4.3****SSD\_MSG\_BOARD\_INFO - Board Information Request****Synopsis**

Message used to retrieve information about the DSI SS7LD Board.

**Format**

MESSAGE HEADER		
Field Name	Meaning	
Type	SSD_MSG_BOARD_INFO (0x7689)	
Id	board_id	
Src	Sending module ID	
Dst	SSD_module_ID	
rsp_req	Used to request a confirmation.	
Hclass	0	
Status	0	
err_info	0	
Len	38	
PARAMETER AREA		
Offset	Size	Name
0	4	ssdmode
4	2	btype
6	4	addrsw
10	2	numsp
12	4	subsysid
16	20	sernum
36	1	cur_temp

37	1	max_temp
----	---	----------

### Description

This message is used when a user application wants to obtain information about a DSI SS7LD Board. This can happen at any time after the board has been reported as being present in the system. Typically, in PCI address mode (see `ssd_mode` below), this message may be sent by the user application to the `ssdl` module at system startup to determine the serial numbers of boards present within the system.

### Parameters

#### **board\_id**

The `board_id` should be set to the logical board number, or, alternatively, if geographic addressing is enabled, to the board's physical address as set using the on-board ADDR switch (SW1).

#### **ssdmode**

Specifies the geographic address mode in which the `ssdl` module is running. This was specified at system start-up.

The geographic address modes values are:

- 1: PCI address mode
- 3: Switch address mode

#### **btype**

The board type. For DSI SS7LD Boards, this parameter is set to 32.

#### **addrsw**

Value of the boards' ADDR switch (SW1) as recorded at power on reset.

#### **numsp**

Number of signaling processors present on the board. For SS7LD boards this will always be 1.

#### **subsysid**

The boards subsystem id.

#### **sernum**

The serial number of the board.

#### **cur\_temp**

Signed 8-bit value containing the current temperature of the board within the range -128 to 127 degrees Celsius. Not supported for SS7LD value will be returned as 0.

#### **max\_temp**

Signed 8-bit value containing the maximum temperature the board has reached since `ssdl` was last started. Value is within the range -128 to 127 degrees Celsius. Not supported for SS7LD value will be returned as 0.

#### 4.4.4 MGT\_MSG\_CONFIG0 - Board Configuration Request

##### Synopsis

Message sent to a board immediately after starting the code running to provide physical configuration parameters.

Note: When using the s7\_mgt protocol configuration utility, this message is generated by s7\_mgt and should not be generated by the user.

##### Format

MESSAGE HEADER		
Field Name	Meaning	
type	MGT_MSG_CONFIG0 (0x7f10)	
id	0	
src	Sending module ID	
dst	MGMT_module_ID	
rsp_req	Used to request a confirmation.	
hclass	0	
status	0	
err_info	0	
len	68	
PARAMETER AREA		
Offset	Size	Name
0	2	config_type
2	2	flags
4	2	l1_flags
6	62	Reserved. Must be set to 0.

##### Description

This message must be the first message sent to the board once the SS7 software is running. It is used to configure layer1 modules on the board for operation. The message contains flags to permit various level 1 configurations. The physical link parameters are configured on a per link basis using the MGT\_MSG\_L1\_CONFIG message.

The confirmation message (if requested) indicates success with a **status** value of 0. To verify that configuration is complete before subsequent messages are issued to the board, the user should request a confirmation message and check the status for success.

If the board is not licensed to run the requested software configuration, a **status** value of 0xfe is returned.

##### Parameters

###### config\_type

Set to 3 when using a DSI SS7LD Board. A separate link layer configuration message should be sent for each link using MGT\_MSG\_L1\_CONFIG message.

**flags**

Global flags with the following bit significance:

- Bit 15 is set to 1 for diagnostics purposes to cause the results of board configuration to be passed to the host. When set, all confirmation messages generated internally on the board during the configuration sequence are sent to the 0xdf **module ID** on the host.
- All other bits are reserved for future use and should be set to 0.

**I1\_flags**

Level 1 flags with the following bit significance:

- Bit 0 controls the layer 1 clock reference source. If set to 0, the clock is recovered from the onboard oscillator. If set to 1, the clock is recovered from one of the line interfaces. Line interfaces can be individually configured with the LIU\_MSG\_CONFIG message to explicitly be excluded from recovering the clock from the interface.
- Bit 6 and 7 together select the initial CT bus clocking mode as shown in the following table. The clocking mode can be modified subsequently and dynamically using the MVD\_MSG\_CNFCLOCK message.

Bit 7	Bit 6	CT bus clocking mode
0	0	<b>The CT bus interface is disabled</b> - The board is electrically isolated from the other boards using the CT bus. The on-board clocks are synchronized to the source selected by bit 0 of this flags parameter.
0	1	<b>Primary Master, A Channel</b> - The board drives CT bus clock set A using the clock source selected by bit 0 of this flags parameter.
1	0	<b>Secondary Master, B Channel</b> - The board is configured to drive clock set B in Secondary Master mode. The on-board clocks are synchronized to the CT bus clock set A. It will automatically switch to become Primary Master if the board driving clock set A fails.
1	1	<b>Slave, initially A Channel</b> – The board uses the CT bus clocks, which must be generated by another board on the CT bus. Initially the board recovers from clock set A, though will switch over automatically to recover from clock set B if set A fails.

- Bit 13 is set to 1 to cause the board to drive the CT\_NETREF1 clocks on the CT bus. The highest priority in-sync line interface is used as a clock source. If this bit is set to zero then CT\_NETREF1 clock is not driven.
- All other bits are reserved and should be set to 0.

## 4.4.5 MGT\_MSG\_L1\_CONFIG - Layer 1 Configuration Request

### Synopsis

Message sent to a board after successful processing of the MGT\_MSG\_CONFIG0 message to configure the layer 1 links.

Note: When using the s7\_mgt protocol configuration utility, this message is generated by s7\_mgt and should not be generated by the user.

### Format

MESSAGE HEADER		
Field Name	Meaning	
type	MGT_MSG_L1_CONFIG (0x7f17)	
id	0	
src	Sending module's module ID	
dst	MGMT_module_ID	
rsp_req	Used to request a confirmation.	
hclass	0	
status	0	
err_info	0	
len	40	
PARAMETER AREA		
Offset	Size	Name
0	2	Reserved. Set to 0.
2	2	l1_resource_id
4	2	data_rate
6	2	link_source
8	4	Reserved. Set to 0.
12	2	link_stream
14	2	link_timeslot
16	4	Reserved. Set to 0.
20	4	options
24	16	Reserved. Set to 0.

### Description

This message is used after successful processing of the MGT\_MSG\_CONFIG0 message to configure physical signaling links. It should only be sent after the MGT\_MSG\_CONFIG0 message has been sent. The message should be sent once for each signaling link to be configured.

### Parameters

Configure the LSL timeslot rate:

#### **l1\_resource\_id**

Layer 1 (logical) resource identifier.

**data\_rate**

Used for setting the link operation. The following table shows the permitted values and their meaning.

Value	Data Rate
0	64 kbits/s
1	56 kbits/s
2	48 kbits/s

**link\_source**

Configure the signaling source.

Set to 0 for DSI SS7LD Board.

**link\_stream**

Signaling stream. This parameter is the physical identity of the T1/E1 line interface containing the signaling link. The value range is 0 to one less than the number of LIUs.

**link\_timeslot**

Signaling timeslot. This field is used to configure conventional SS7 links. The valid range is 1 to 24 for a T1/J1 interface and 1 to 31 for an E1 interface.

**options**

A 32-bit value containing run-time options as follows:

- **Bit 0** - Set to 1 to disable automatic FISU generation. This is normally required for Japanese MTP operation only.
- **All Other Bits** - Must be set to 0.

## 4.4.6 MGT\_MSG\_L1\_END - Layer 1 Configuration End

### Synopsis

Message sent to a board to remove an existing layer 1 link that was previously configured by sending an MGT\_MSG\_L1\_CONFIG message.

### Format

MESSAGE HEADER		
Field Name		Meaning
type		MGT_MSG_L1_END (0x7f18)
id		0
src		Sending module's module ID
dst		MGMT_module_ID
rsp_req		Used to request a confirmation.
hclass		0
status		0
err_info		0
len		4
PARAMETER AREA		
Offset	Size	Name
0	2	Reserved. Must be set to 0.
2	2	l1_resource_id

### Parameters

The MGT\_MSG\_L1\_END message includes the following parameter:

#### **l1\_resource\_id**

Layer 1 (logical) resource identifier.



## 4.5 Hardware Control Messages

Hardware control messages are used to control various hardware devices on the board, including the T1/E1 Line Interface Units (LIUs and the clocking mode for the board.

In a static configuration, these hardware blocks can be set up using the `s7_mgt` protocol configuration utility along with the appropriate commands in the `config.txt` protocol configuration file.

If dynamic control of the hardware is required (or the user has elected not to use `s7_mgt`), the user application must build and send at least some of the hardware control messages.

The messages in the hardware control category include:

- `LIU_MSG_CONFIG` - LIU Configuration Request
- `LIU_MSG_CONTROL` - LIU Control Request
- `LIU_MSG_R_CONFIG` - LIU Read Configuration Request
- `LIU_MSG_R_CONTROL` - LIU Read Control Request
- `MVD_MSG_CNFCLOCK` - Configure Clock Request
- `MVD_MSG_CLOCK_PRI` - Configure Clock Priority Request
- `MVD_MSG_RESETSWX` - Reset Switch Request
- `MVD_MSG_SC_CONNECT` - Connect Request

### 4.5.1 LIU\_MSG\_CONFIG - LIU Configuration Request

#### Synopsis

Message sent by the application to establish the operating mode for a Line Interface Unit (LIU).

Note: When using the s7\_mgt protocol configuration utility, this message is generated by s7\_mgt as a result of the LIU\_CONFIG command. It therefore need not be generated by the user.

#### Format

MESSAGE HEADER		
Field Name	Meaning	
type	LIU_MSG_CONFIG (0x7e34)	
id	liu_id (in the range 0 to one less than the number of LIUs)	
src	Sending module ID	
dst	MVD_module_ID	
rsp_req	Used to request a confirmation.	
hclass	0	
status	0	
err_info	0	
len	40	
PARAMETER AREA		
Offset	Size	Name
0	1	liu_type
1	1	liu_code
2	1	frame_format
3	1	crc_mode
4	1	build_out
5	6	Reserved. Must be set to 0.
11	1	ais_gen
12	6	Reserved. Must be set to 0.
18	1	sensitivity
19	2	clk_opt
21	19	Reserved. Must be set to 0.

#### Description

This message is sent to the board to configure the operating mode of a Line Interface Unit (LIU). All configuration parameters must be supplied in the message, that is, it is not possible to modify individual operating parameters in isolation. On receipt of the message, the board first verifies that the fitted hardware options support the requested operating mode and then initializes (or reinitializes) the LIU.

The confirmation message (if requested) indicates success with a **status** value of 0.

## Parameters

A description of the permitted parameter values are given below. When the board is initially configured, the LIUs are initialized to a disabled condition.

### liu\_type

The physical interface type according to the following table:

Value	Description
1	Disabled (used to deactivate a LIU). In this mode, the LIU does not produce an output signal.
3	E1 120 ohm balanced interface
4	T1
5	E1 – Impedance based on hardware (The SS7LD board supports only 120 ohm balanced interfaces for E1 but this value is supported for compatibility with other boards)

### line\_code

The line coding technique. The following table shows the permitted values and their meanings.

Value	Description
1	HDB3 (E1 only)
2	AMI
4	B8ZS (T1)

### frame\_format

The frame format. The following table shows the permitted values and their meanings.

Value	Description
1	E1 double frame (E1 only)
2	E1 CRC4 multiframe (E1 only)
4	D3/D4 (Yellow alarm = bit 2 in each channel (T1 only)
7	ESF (Yellow alarm in data link channel) (T1 only)

### crc\_mode

The CRC mode. The following table shows the permitted values and their meanings.

Value	Description
1	CRC generation disabled
2	CRC4 enabled (frame_format must be set to 2)
4	CRC6 enabled (frame_format must be set to 7)

**build\_out**

The following table shows the permitted values and their meanings.

Value	Description
0	Setting for E1 devices
1	T1 default (T1 short haul)
2	T1 short haul 0 - 110 ft
3	T1 short haul 110 – 220 ft
4	T1 short haul 220 – 330 ft
5	T1 short haul 330 – 440 ft
6	T1 short haul 440 – 550 ft
7	T1 short haul 550 – 600 ft
8	T1 long haul LB0 (-0dB)

**ais\_gen**

The (initial) mode used to generate the Alarm Indication Signal (Blue alarm). The user may subsequently modify the setting of the outgoing signal using the LIU\_MSG\_CONTROL message. The following table shows the permitted values and their meanings.

Value	Description
1	Disabled; do not generate AIS/Blue alarm
2	Enabled; generate AIS/Blue alarm

**sensitivity**

The mode settings to configure the interface sensitivity for monitoring purposes. The following table shows the permitted values and their meanings.

Value	Description
1	Terminated; normal impedance presented on the line
2	High impedance mode; set the LIU to high impedance for monitoring.
4	PMP mode; set the LIU sensitivity to operate with a Protected Monitoring Point.

### clk\_options

A 16-bit value containing clocking options for the LIU. This value provides the ability to override default LIU clocking options for each LIU. Default options are specified per board within the MGT\_MSG\_CONFIG0 message.

- Bit 0 - Disable LIU clock recovery for this interface.
- All other bits set to 0.

## 4.5.2 LIU\_MSG\_CONTROL - LIU Control Request

### Synopsis

Message sent by the application to dynamically control operation for a Line Interface Unit (LIU). Allows setting of outgoing alarms and diagnostic loopbacks.

### Format

MESSAGE HEADER		
Field Name	Meaning	
type	LIU_MSG_CONTROL (0x7e35)	
id	liu_id (in the range 0 to one less than the number of LIUs)	
src	Sending module ID	
dst	MVD_module_ID	
rsp_req	Used to request a confirmation.	
hclass	0	
status	0	
err_info	0	
len	16	
PARAMETER AREA		
Offset	Size	Name
0	1	ais_gen
1	1	Reserved for future use, must be set to 0
2	1	loop_mode
3	13	Reserved for future use, must be set to 0.

### Description

This message is sent to the board to perform dynamic changes to the operation of the Line Interface Unit (LIU). It allows the user to control the generation of AIS (Blue alarm) and to activate various diagnostic loopback modes.

The confirmation message (if requested) indicates success with a **status** value of 0.

**Parameters****ais\_gen**

The mode used to generate the Alarm Indication Signal (Blue alarm). The following table shows the permitted values and their meanings.

Value	Description
0	Do not change AIS/Blue alarm generation mode
1	Disabled; do not generate AIS/Blue alarm
2	Enabled; generate AIS/Blue alarm

**loop\_mode**

The diagnostic *loopback* mode. The following table shows the permitted values and their meanings.

Value	Description
0	Do not change diagnostic loopback mode
1	Disabled - remove any diagnostic loop
2	Payload loopback
4	Local loopback

**4.5.3 LIU\_MSG\_R\_CONFIG - LIU Read Configuration Request****Synopsis**

Message sent by the application to read back the current LIU configuration from the board.

**Format**

MESSAGE HEADER		
Field Name	Meaning	
type	LIU_MSG_R_CONFIG (0x5e37)	
id	liu_id (in the range 0 to one less than the number of LIUs)	
src	Sending module ID	
dst	MVD_module_ID	
rsp_req	Used to request a confirmation.	
hclass	0	
status	0	
err_info	0	
len	40	
PARAMETER AREA		
Offset	Size	Name
0	40	Parameter area formatted in the same manner as the LIU_MSG_CONFIG message. All fields should be set to 0. The confirmation message contains the board configuration details. The user should set the fields to 0 and the module writes the current configuration parameters in the confirmation message.

**Description**

This message is sent to the board to read back the current operating configuration of the Line Interface Unit (LIU). The user should always request a confirmation message. The confirmation message indicates success with a **status** value of 0 and contains the current configuration parameters in the parameter area of the message.

**4.5.4 LIU\_MSG\_R\_CONTROL - LIU Read Control Request****Synopsis**

Message sent by the application to read back the current Line Interface Unit (LIU) control options from the board.

**Format**

MESSAGE HEADER		
Field Name	Meaning	
type	LIU_MSG_R_CONTROL (0x5e38)	
id	liu_id (in the range 0 to one less than the number of LIUs)	
src	Sending module ID	
dst	MVD_module_ID	
rsp_req	Used to request a confirmation.	
hclass	0	
status	0	
err_info	0	
len	16	
PARAMETER AREA		
Offset	Size	Name
0	16	Parameter area formatted in the same manner as the LIU_MSG_CONTROL message. All fields should be set to 0. The confirmation message contains LIU control options. The user should set the fields to 0 and the module writes the current control parameters in the confirmation message.

**Description**

This message is sent to the board to read back the current control parameters selected for a Line Interface Unit (LIU). The user should always request a confirmation message. The confirmation message indicates success when the **status** value of 0 and contains the current control parameters in the parameter area of the message.



## 4.5.5 MVD\_MSG\_CNFCLOCK - Configure Clock Request

### Synopsis

Message sent to a DSI SS7LD Board to configure the clocking mode for the board.

### Format

MESSAGE HEADER		
Field Name	Meaning	
type	MVD_MSG_CNFCLOCK (0x7e20)	
id	0	
src	Sending Module ID	
dst	MVD_TASK_ID	
rsp_req	used to request a confirmation	
hclass	0	
status	Status Response (if confirmation requested)	
err_info	0	
len	12	
PARAMETER AREA		
Offset	Size	Name
0	2	bus_speed
2	2	clk_mode
4	2	pll_clk_src
6	2	ref1_mode
8	4	Reserved. Set to zero

### Description

This message is used to control the on-board clock circuitry. It allows the user to select the CT bus clocking mode and the reference clock sources for the local and bus reference clocks.

The confirmation message (if requested) indicates success by status of zero.

### Parameters

#### bus\_speed

This parameter is used to set the CT bus speed; the permissible values are as follows:

Value	Bus speed
0	No change
3	8.192 MHz

#### clk\_mode

This parameter determines the clocking mode for the DSI SS7LD Board; the permissible values are as follows:

Value	Clock Mode
0	No change

1	CT bus Primary Master, driving Clock Set A
2	CT bus Secondary Master, driving Clock Set B
3	CT bus Slave, initially using Clock Set A
4	CT bus disabled
10	CT bus Primary Master, driving Clock Set B
11	CT bus Secondary Master, driving Clock Set A
12	CT bus Slave, initially using Clock Set B

When mode 4 is selected ("CT bus disabled"), the DSI SS7LD Board is electrically isolated from the other boards using the CT bus. The on-board clocks are synchronized to the configured `pll_clk_src` reference.

If the DSI SS7LD Board is configured to be Slave to the CT bus, then it automatically switches between using Clock Set A and Clock Set B if it detects a failure on the current clock set.

When a board is acting as Primary Master, it uses the clock reference set by the `pll_clk_src` parameter to drive the CT bus clock.

As Secondary Master, the `pll_clk_src` must be set to an appropriate source ready for use if the board acting as Primary Master stops driving the CT bus clock. Until this time, the on-board clocks on the Secondary Master board are synchronized to the CT bus clock provided by the Primary Master.

### **pll\_clk\_src**

This parameter determines the source of the PLL reference clock, the permissible values are:

Value	PLL Clock Source
0	No change
1	Clock recovered from one of the line interfaces according to priority order.
5	Local reference oscillator
7	NETREF 1

The PLL clock is used as the reference when acting as CT bus Primary Master.

If the clock is to be recovered from one of the line interfaces, then the highest-priority in sync line interface is used as the reference. Each line interface is assigned a priority: by default `liu_id=0` is the highest priority and `liu_id=7` the lowest. The user may modify the priority order by sending the `MVD_MSG_CLOCK_PRI` message. If none of the interfaces are available for recovery, then the phase locked loop runs in holdover mode, outputting a clock with the same frequency as the last valid signal. When a valid signal returns, it waits for a short period to verify that it is stable and then automatically switches to use it as the clock reference.

If using one of the NETREF signals as the reference source, then another board in the system must be providing this reference by driving a clock source onto the appropriate CT bus NETREF lines. If the NETREF signal is lost, the board continues with the PLL in holdover mode until another `MVD_MSG_CNFCLOCK` message is received to switch to a new mode.

Note: If the NETREF signal recovers, it is still necessary to re-set the clock configuration and move out of holdover mode by sending MVD\_MSG\_CNFCLOCK and re-selecting the appropriate mode.

### ref1\_mode

This parameter determines whether the CT bus NETREF\_1 clock is driven onto the CT bus by this board. The permissible values are as follows:

Value	NETREF_1 clock Mode
0	No Change
1	Drive NETREF_1 using clock recovered from highest priority line interface.
6	Tri-state (i.e., Not driven)

When the NETREF\_1 signal is being driven then the clock source is the highest priority line interface. If no interface is available for clock recovery, then no signal is driven onto the bus.

Driving the NETREF\_1 signal is independent of the clk\_mode and pll\_clk\_src settings for this board.

### Status Response

The confirmation message (if requested) indicates success by status of zero.

On error, the following status value can be found in the confirmation message.

Value	Mnemonic	Description
0xff	None	Request to configure clocking mode fails.

## 4.5.6 MVD\_MSG\_CLOCK\_PRI - Configure Clock Priority Request

### Synopsis

Message sent to SS7LD Board to configure the clock recovery priority order.

### Format

MESSAGE HEADER		
Field Name	Meaning	
type	MVD_MSG_CLOCK_PRI (0x7e21)	
id	0	
src	Sending Module ID	
dst	MVD_TASK_ID	
rsp_req	used to request a confirmation	
hclass	0	
status	Status Response (if confirmation requested)	
err_info	0	
len	16	
PARAMETER AREA		
Offset	Size	Name
0	1	liu0_pri
1	1	liu1_pri
2	1	liu2_pri
3	1	liu3_pri
4	12	Reserved, set to 0.

### Description

This message allows the user to specify a priority for each line interface. When configured to recover clock from the line interfaces, this priority is used to decide which line interface to use as the clock source. The highest priority in-sync line interface is used, with the board automatically moving through the list of clock sources as line interfaces lose synchronization or are deemed stable again. If no interfaces are in sync, the board remains in "holdover" mode, based on the last valid clock that was recovered.

The confirmation message (if requested) indicates success by status of zero.

### Parameters

#### liu<sub>n</sub>\_pri

The relative priority for each LIU using the values taken from the following table:

Value	Meaning
0	No change to the interface's priority.
1 ... 32	New priority value for the line interface. The value 1 indicates highest priority, 32 the lowest priority. If two interfaces are given the same priority, the lowest-numbered interface is used first.
255	Special value indicating that the line interface must not be used for clock recovery.

### Status Response

The confirmation message (if requested) indicates success by status of zero.

On error, the following status value can be found in the confirmation message.

Value	Mnemonic	Description
0xff	None	Request to configure clock recovery priority order fails.

## 4.5.7

### MVD\_MSG\_RESETSWX - Reset Switch Request

#### Synopsis

Resets the digital switch to its default state in accordance with the current board configuration.

#### Format

MESSAGE HEADER	
Field Name	Meaning
type	MVD_MSG_RESETSWX (0x7e00)
id	0
src	Sending module ID
dst	MVD_module_ID
rsp_req	Used to request a confirmation.
hclass	0
status	0
err_info	0
len	0

#### Description

This message is sent to the board to reset the state of the digital cross connect switch.

The confirmation message (if requested) indicates success with a status value of 0. On receipt of the confirmation message, the operation to reset the switch is completed.

## 4.5.8 MVD\_MSG\_SC\_CONNECT - Connect Request

### Synopsis

Message sent to the board to control the switch path.

### Format

MESSAGE HEADER		
Field Name	Meaning	
type	MVD_MSG_SC_CONNECT (0x7e1f)	
id	0	
src	Sending module ID	
dst	MVD_module_ID	
rsp_req	Used to request a confirmation.	
hclass	0	
status	0	
err_info	0	
len	16	
PARAMETER AREA		
Offset	Size	Name
0	2	local_stream
2	2	local_slot
4	2	mode
6	2	source_stream
8	2	source_slot
10	4	Reserved. Must be set to 0.
14	2	pattern

### Description

This message is sent to the board to control the cross connect switch. Several different actions can be performed depending on the value of the mode parameter. These include:

- Transmission of a fixed pattern on a specified timeslot
- Initialising a simplex connection between two LIUs
- Initialising a duplex connection between two LIUs.
- Removal of duplex connection between two LIUs

Attempting to use this message in a run mode where the cross connect switch is disabled will result in a failure return code.

The confirmation message (if requested) indicates success with a status value of 0.

### Parameters

The parameters that can be included in the MVD\_MSG\_SC\_CONNECT message depend on the requested mode. The following table depicts the parameters that are required for each mode:

Mode	Required Parameters				
	local_stream	local_slot	source_stream	source_timeslot	pattern
10	Yes	Yes	0	0	Yes
11	Yes	Yes	Yes	Yes	0
12	Yes	Yes	Yes	Yes	0
13	Yes	Yes	Yes	Yes	0

**local\_stream**

Defines which local stream to use. The local\_stream parameter specifies the T1/E1 Line Interface Unit (LIU) to be used as the output. Values for the LIU are in the range 0 to one less than the number of LIUs supported.

**local\_slot**

Defines which local\_stream timeslot to use. The valid range is 1 to 24 for a T1/J1 interface and 1 to 31 for an E1 interface.

**mode**

Determines the operating mode. The following table shows the permitted values and their meanings.

Value	Meaning
10	Generate a fixed pattern, for example an Idle pattern, on a local timeslot. The local_stream parameter specifies the liu_id, the local_slot parameter specifies the timeslot and the pattern parameter specifies the 8-bit data to be output on the timeslot.
11	Make a simplex connection between two local bus timeslots. The source_stream and source_slot parameters specify the source of the signal in terms of liu_id and timeslot, respectively. The local_stream and local_slot parameters specify the outgoing liu_id and timeslot, respectively.
12	Make a duplex connection between two local bus timeslots. The source_stream and source_slot parameters specify the source of the signal in terms of liu_id and timeslot, respectively. The local_stream and local_slot parameters specify the outgoing liu_id and timeslot, respectively.
13	Remove a duplex connection between two local bus timeslots. Use the local_stream and local_slot parameters to specify one timeslot and the source_stream and source_slot parameters to specify the other.

**source\_stream**

The source\_stream parameter specifies the T1/E1 Line Interface Unit (LIU) to be used as the source. Values for the LIU are in the range 0 to one less than the number of LIUs supported.

**source\_slot**

Defines which source\_stream timeslot to use. The valid range is 1 to 24 for a T1/J1 interface and 1 to 31 for an E1 interface.

**pattern**

The value of the fixed data to be generated. The value must be in the range 0 to 255. Typical values are 0xff for an "all ones" idle pattern, or 0x2a for an ITU-T E1 idle pattern.



## 4.6 Signaling Interface Messages

Signaling interface messages allow signaling links to be activated and deactivated by the user and provide a mechanism for communication between the MTP3 module and the user part module (for example, ISUP, TUP or SCCP). In many cases, the user part module is a Dialogic® DSI Protocol Stack so the user does *not* need to handle the MTP primitives as they pass directly between MTP3 and the user part module.

In the case that the user application is implementing the user part functionality, the MTP primitives are applicable. See the *Dialogic® SS7 Protocols MTP2 Programmer's Manual* and the *Dialogic® SS7 Protocols MTP3 Programmer's Manual* for more information.

## 4.7 **Event Indication Messages**

Event indication messages are the mechanism by which protocol and software error events are reported to the application. These messages are generated asynchronously by different modules within the stack.

The messages in the event indication category include:

- SSD\_MSG\_STATE\_IND - Board Status Indication
- MVD\_MSG\_CLK\_IND - Clock Event Indication
- MVD\_MSG\_LIU\_STATUS - LIU Status Indication
- DVR\_MSG\_TRACE\_MASK - Set Trace Mask Request

## 4.7.1 SSD\_MSG\_STATE\_IND - Board Status Indication

### Synopsis

Message sent to the application on completion of the reset and download sequence or on detection of a board status event.

### Format

MESSAGE HEADER		
Field Name	Meaning	
type	SSD_MSG_STATE_IND (0x06a0)	
id	board_id	
src	SSD_module_ID (0x20)	
dst	mgmt_id for SSD	
rsp_req	Used to request a confirmation	
hclass	0	
status	event_type (see below)	
err_info	0	
len	4	
PARAMETER AREA		
Offset	Size	Name
0	2	board_type
2	2	failure_code

### Description

This message is used to convey the status of a board reset operation (success or failure) to the user. The status is indicated in the status field of the message header. The following table shows the possible event\_type values:

### event\_type

Value	Meaning
0x60	Reset successful
0x62	Board failure
0x66	License validation failure
0x67	License appears corrupt
0x70	Message congestion toward board cleared
0x71	Message congestion toward board onset

### Parameters

#### board\_type

Set to 32 for SS7LD.

**failure\_code**

Value	Meaning
0x0000	undefined
0x00d7	Thermal failure

**4.7.2 MVD\_MSG\_CLK\_IND - Clock Event Indication****Synopsis**

Message issued by the board to indicate on-board clocking related events.

**Format**

MESSAGE HEADER	
Field Name	Meaning
type	MVD_MSG_CLK_IND (0x0e23)
id	0
src	MVD_TASK_ID
dst	0xdf
rsp_req	0
hclass	0
status	event_id
err_info	0
len	0

**Description**

This message is issued by the board to indicate events within the on-board clocking circuitry.

**event\_id**

This field specifies the event that caused the indication to be generated:

event_id	Description
1	<b>PLL entered hold-over mode</b> Issued by boards acting as primary or secondary clock master when its nominated clock reference becomes unavailable. The phase-locked-loop starts operating in "hold-over" mode, continuing to generate an on-board clock at the same frequency as the last valid reference signal.
2	<b>PLL left hold-over mode</b> The nominated clock reference for a primary or secondary master board has become available and the is now being used as the input to the board's clock circuitry.
3	<b>CT bus clock set A fail</b> The CT bus clock set A signals are not being correctly driven.
4	<b>CT bus clock set A recover</b> The CT bus clock set A signals are being driven.
5	<b>CT bus clock set B fail</b> The CT bus clock set B signals are not being correctly driven.
6	<b>CT bus clock set B recover</b> The CT bus clock set B signals are being driven.

event_id	Description
7	<b>Master clock changeover</b> The board issuing this indication has automatically changed from secondary master to primary master role for the clock set it was configured to drive.

### 4.7.3 MVD\_MSG\_LIU\_STATUS - LIU Status Indication

#### Synopsis

Message issued by the board to provide notification of changes in LIU status.

#### Format

MESSAGE HEADER	
Field Name	Meaning
type	MVD_MSG_LIU_STATUS (0x0e01)
id	liu_id (in the range 0 to one less than the number of LIUs)
src	MVD_module_ID
dst	MGMT_module_ID
rsp_req	0
hclass	0
status	liu_status (see below)
err_info	Reserved for future use.
len	0

#### Description

This message is issued by the board for every change of state on the trunk interface.

#### Parameters

##### liu\_id

The identity of the Line Interface Unit (LIU) to which the status indication applies.

##### liu\_status

The LIU status. The following table shows the possible values and their meanings.

Value	Mnemonic	State
10	LIUS_SYNC_LOSS	Frame Sync Loss
11	LIUS_IN_SYNC	Frame Sync OK
12	LIUS_AIS	AIS Detected
13	LIUS_AIS_CLRD	AIS Cleared
14	LIUS_REM_ALARM	Remote Alarm
15	LIUS_REM_ALM_CLRD	Remote Alarm Cleared
20	LIUS_PCM_LOSS	PCM Loss
21	LIUS_PCM_OK	PCM Restored

#### 4.7.4 DVR\_MSG\_TRACE\_MASK - Set Trace Mask Request

##### Synopsis

Message issued to HDLC driver to cause per-link tracing of protocol primitives.

##### Format

MESSAGE HEADER		
Field Name	Meaning	
type	DVR_MSG_TRACE_MASK (0x731b)	
id	lower layer link id	
src	Originating module_id	
dst	DVR_ALT_TASK_ID	
rsp_req	Sending layer's bit set if response required	
hclass	0	
status	0	
err_info	0	
len	12	
PARAMETER AREA		
Offset	Size	Name
0	4	op_evt_mask – Output event trace mask
4	4	ip_evt_mask – Input event trace mask
8	4	Reserved set to 0.

## Description

The driver module supports comprehensive tracing options on a per-link and per-primitive basis. The module can be configured to trace any message received or transmitted and a number of management events. This message is used to selectively enable tracing of events. It can be used at any time during operation and continues to be effective until the next Trace Mask Set Request is received for the same link.

Traced events are indicated to the management module using the MGT\_MSG\_TRACE\_EV Event Indication.

## Parameters

### op\_evt\_mask

The output event trace mask. This is a 32-bit value with bits set to 1 to cause a trace message to be sent to the management module whenever a message is issued by the driver. Care should be taken when tracing messages because the system throughput may be reduced. The fields in the trace mask cause the events indicated in the following table to be traced.

Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	SUERM FAIL	AERM FAIL	RDY	BUSY IND		RX IND	
Key:				<ul style="list-style-type: none"> <li>• SUERM FAIL – SUERM Failure Indication</li> <li>• AERM FAIL - AERM Failure Indication</li> <li>• RDY – Ready</li> <li>• BUSY IND – Busy indication</li> </ul>			
				<ul style="list-style-type: none"> <li>• RX IND – Receive message indication</li> </ul>			
Note: The shaded boxes relate to internal events within the board and are of limited use to the user.							

**ip\_evt\_mask**

The input event trace mask. This is a 32-bit value with bits set to 1 to cause a trace message to be sent to the management module whenever a message is received by the driver. Care should be taken when tracing messages as system throughput is reduced. The fields in the trace mask cause the events indicated in the following table to be traced.

Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
						SUERM STOP	SUERM START
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AERM STOP	AERM START	REP NEXT SU	RTP		TM EXP		TX REQ
Key: <ul style="list-style-type: none"> <li>• SUERM STOP – SUERM Stop Request</li> <li>• SUERM START – Start SUERM Request</li> <li>• AERM STOP – AERM Stop Request</li> <li>• AERM START – Start AERM Request</li> <li>• REP NEXT SU – Reports next SU received</li> </ul>				<ul style="list-style-type: none"> <li>• RTP – Return to pool</li> <li>• TM EXP – Timer expiry indication</li> <li>• TX REQ – Transmit request</li> </ul>			
Note: The shaded boxes relate to internal events within the board and are of limited use to the user.							



## 4.8 Status Request Messages

Status request messages can be used to poll the status of modules or systems running on the board.

The messages in the status request category include:

- LIU\_MSG\_R\_STATE - LIU Read State Request
- LIU\_MSG\_R\_STATS - LIU Read Statistics Request
- MVD\_MSG\_R\_CLK\_STATUS - Clock Status Request
- MGT\_MSG\_R\_BRDINFO - Read Board Info Request
- DVR\_MSG\_R\_L1\_STATS - Link Statistics Request

### 4.8.1 LIU\_MSG\_R\_STATE – LIU Read State Request

#### Synopsis

Message sent by the application to read the current state of a Line Interface Unit (LIU).

#### Format

MESSAGE HEADER		
Field Name	Meaning	
type	LIU_MSG_R_STATE (0x5e39)	
id	liu_id (in the range 0 to one less than the number of LIUs)	
src	Sending module ID	
dst	MVD_module_ID	
rsp_req	Used to request a confirmation.	
hclass	0	
status	0	
err_info	0	
len	1	
PARAMETER AREA		
Offset	Size	Name
1	1	state

#### Description

This message is sent to the board to read the current operating state of a Line Interface Unit (LIU). The user should always request a confirmation message. The confirmation message indicates success with a **status** value of 0 and contains the current LIU state information in the parameter area of the message.

#### Parameters

##### state

The current state of the LIU as shown in the following table.

Value	Description
0	OK
1	PCM Loss
2	AIS
3	Sync Loss
4	Remote Alarm

## 4.8.2 LIU\_MSG\_R\_STATS - LIU Read Statistics Request

### Synopsis

Message used to read back performance statistics associated with a Line Interface Unit (LIU).

### Format

MESSAGE HEADER		
Field Name	Meaning	
type	LIU_MSG_R_STATS (0x5e36)	
id	liu_id (in the range 0 to one less than the number of LIUs)	
src	Sending module ID	
dst	MVD_module_ID	
rsp_req	Used to request a confirmation.	
hclass	0	
status	0 to read statistics 1 to read statistics and reset counters	
err_info	0	
len	42	
PARAMETER AREA		
Offset	Size	Name
0	2	version
2	2	Reserved. Must be set to 0.
4	4	duration
8	4	bit_errors
12	4	code_violations
16	4	frame_slips
20	4	oos_transitions
24	4	errored_seconds
28	4	severely_errored_seconds
32	2	prbs_status
34	4	Reserved. Must be set to 0.
38	4	Reserved. Must be set to 0.

### Description

This message is used to collect performance statistics for a given Line Interface Unit (LIU). A module requesting LIU statistic information is required to complete the version parameter of the message, request a response, and set all additional parameter values to zero.

The confirmation message shall feature a non-zero status in the event of an error. In the event of successful retrieval of information, the message parameter field shall contain LIU information as specified in the message format. The statistics can either be read and left unchanged, or read and reset in a single operation depending on the setting of the status field in the request message.

Typically, a managing application would be set up to periodically (for example, hourly or daily) read and reset the statistics and store the resulting information so that it can be accessed later for generation of performance reports for the line interface.

## Parameters

### version

Version of the parameter area.

### duration

The duration (in seconds) since the statistics were last reset.

### bit\_errors

A count of the actual number of bit errors detected by the framer device for the LIU. The precise meaning of this parameter varies depending on the operating mode of the framer:

- For E1 operating modes, it is the number of errors detected in the frame alignment word.
- For T1 interfaces operating in D3/D4 frame format, it is the number of framing bit errors.
- For T1 interfaces operating in ESF format, it is the number of CRC6 errors.

Note: In general, the user should use the `errored_seconds` and `severely_errored_seconds` parameters instead since these parameters provide normalized values that have the same meaning for all modes of operation.

### code\_violations

A count of all the line code violations detected on the interface.

### frame\_slips

A count of the number of frame slips that have occurred on the interface.

### oos\_transitions

A count of the number of transitions from the in synchronization state to the out of synchronization state.

### errored\_seconds

The number of seconds since the statistics were last reset during which the interface contained errors. An *errored second* is any second during which the interface is out of synchronization, or there are frame slips, bit errors, or line code violations.

### severely\_errored\_seconds

The number of severely errored seconds since the statistics were last reset. A *severely errored second* is

a second during which the interface is out of synchronization or the bit error rate exceeds 1 in 1,000.

### prbs\_status

The status of Pseudo Random Bit Sequence (PRBS) indications.

- 1 = PRBS is valid, the counts are correct.
- 3 = PRBS sequence is not synchronized.

### 4.8.3 MVD\_MSG\_R\_CLK\_STATUS – Clock Status Request

#### Synopsis

Message sent by the application to read the current clocking state.

#### Format

MESSAGE HEADER		
Field Name	Meaning	
type	MVD_MSG_R_CLK_STATUS (0x6e04)	
id	0	
src	Sending module's Id	
dst	MVD_TASK_ID	
rsp_req	Used to request a confirmation	
hclass	0	
status	0	
err_info	0	
len	10	
PARAMETER AREA		
Offset	Size	Name
0	2	clk_mode
2	2	status_a
4	2	status_b
6	2	pll_state
8	2	liu_rec

#### Description

This message is sent to the board to read the current clocking state. The user should always request a confirmation message. The confirmation message indicates success with a **status** value of 0 and contains the current state in the parameter area of the message.

#### Parameters

##### clk\_mode

Indicates the current clock mode for the board as detailed in the following table.

Value	Clock Mode
1	CT Bus Primary Master, driving clock set A
2	CT Bus Secondary Master, driving clock set B
3	CT Bus Slave, initially using clock set A
4	CT Bus disabled
10	CT Bus Primary Master, driving clock set B
11	CT Bus Secondary Master, driving clock set A

12	CT Bus Slave, initially using clock set B
----	---

**status\_a**

The following table shows the possible values and their meanings.

Value	Description
1	CT clock set A OK
2	CT clock set A fail

**status\_b**

The following table shows the possible values and their meanings.

Value	Description
1	CT clock set B OK
2	CT clock set B fail

**pll\_state**

Value	Description
0	Configured as CT Bus slave, PLL not used.
1	Recovering from LIU specified in liu_rec parameter
5	Local reference oscillator
7	NETREF_1
9	PLL in holdover mode

**liu\_rec**

If the PLL is recovering the clock from a line interface, this parameter indicates the liu\_id (in the range 0 to one less than the number of LIUs) of the LIU being used as a reference. Otherwise, this parameter should be ignored.

#### 4.8.4 MGT\_MSG\_R\_BRDINFO - Read Board Info Request

**Synopsis**

Message used to request basic board information.

**Format**

MESSAGE HEADER		
Field Name	Meaning	
type	MGT_MSG_R_BRDINFO (0x6f0d)	
id	0	
src	Sending module ID	
dst	MGMT_module_ID	
rsp_req	Used to request a confirmation	
hclass	0	
status	0	
err_info	0	
len	60	
PARAMETER AREA		
Offset	Size	Name
0	1	board_type
1	1	board_rev
2	2	Reserved
4	1	swb
5	21	Reserved
26	4	Dram_size
30	20	bsn
50	4	Reserved
54	1	current_temp
55	1	max_temp
56	4	Reserved

**Description**

This message is provided to request a reply indicating the values of a number of attributes associated with the board. On receipt of this request, the module returns the message with the status "SUCCESS - 0" to the sender and includes the information requested.

**Parameters****board\_type**

The board type which is set to 32 for the SS7LD board.

**board\_rev**

The board revision number. Currently 0.

**swb**

Geographic addressing switch setting, that is, the address at which the board will appear when the -o3 feature of sssl is used.

**dram\_size**

DRAM size in bytes.

**bsn**

The board's production serial number (ASCII characters, null terminated).

**current\_temp**

Signed 8-bit value containing the current temperature of the board within the range -128 to 127 degrees Celsius.

**max\_temp**

Signed 8-bit value containing the maximum temperature the board has reached since ssdl was last started. Value is within the range -128 to 127 degrees Celsius.

## 4.8.5 DVR\_MSG\_R\_L1\_STATS - Link Statistics Request

**Synopsis**

Retrieve link statistics.

**Format**

MESSAGE HEADER		
Field Name	Meaning	
type	DVR_MSG_R_L1_STATS (0x6136)	
id	l1_llid	
src	Sending module ID	
dst	module ID of onboard HDLC/SS7 driver	
rsp_req	Used to request a confirmation, sending layer's bit must be set.	
hclass	0	
status	0 – Read statistics 1 – Read statistics and reset	
err_info	0	
len	48	
PARAMETER AREA		
Offset	Size	Name
0	4	duration
4	4	abort_cnt
8	4	CRC_errs
12	4	misaligned_frames.
16	4	length_errs
20	4	rx_overrun
24	4	receiver_busy_cnt
28	4	rx_frame_cnt
32	4	rx_pre_filter_cnt
36	4	tx_frame_cnt
40	4	rx_pool_cnt.
44	4	rx_busy_status

**Description**

This message provides the user with a number of statistics on a per link basis. If the user sends the message with a non zero status field, the statistics are reset to 0 after being read.



## **Parameters**

### **duration**

Duration in tenths of a second since the statistic counters were last reset.

### **about\_cnt**

The number of aborts received on the link.

### **CRC\_errs**

Number of CRC errors received on the link.

### **misaligned\_frames**

The number of misaligned frames (that is, frames that are not an integer multiple of 8 octets) received on the link.

### **length\_errs**

The number of received frames that were designated as either too long or too short for a configured protocol.

### **rx\_overflow**

The number of times that the receiver was forced to discard incoming frames as a result of there being no internal buffers available to receive the incoming data. This is a count of the number of events rather than a count of the number of frames discarded.

### **receiver\_busy\_cnt**

The number of times the receiver has entered the busy state as a result of the number of internal buffers falling below a set threshold.

### **rx\_frame\_cnt**

The number of (error-free) frames received on the link, excluding any duplicate frames that are discarded as a result of the internal filtering mechanism.

### **rx\_pre\_filter\_cnt**

The total number of (error-free) frames received on the link including any duplicate frames that are discarded as a result of the internal filtering mechanism.

### **tx\_frame\_cnt**

The number of frames transmitted on the link excluding any repeated frames that are generated automatically (for example, repeated FISUs or LSSUs).

### **rx\_pool\_cnt**

The number of message buffers currently available in the links internal receive pool.

### **rx\_busy\_status**

Normally set to 0, but in the event of the receiver being in the a "busy" state (where the number of internal buffers falls below a fixed internal threshold), this field is set to 1.

## Appendix A. Protocol Configuration Using Discrete Messages

This appendix provides guidelines for protocol configuration using individual messages.

### A.1 Protocol Configuration Using Individual Messages

As an alternative to using the `s7_mgt` protocol configuration utility it is possible to perform protocol configuration by building and sending messages directly to the board. This approach means that it is necessary to write some application code to handle configuration, but has the advantage that the application can, if required, reconfigure the board without restarting the application.

Communication with the board is achieved by sending and receiving messages. The configuration sequence is described below. The application should allocate a message structure using the `getm( )` library function and send it to the board using the `GCT_send( )` library function. The application should periodically call the `GCT_receive( )` or `GCT_grab( )` library functions to receive messages from the board. The `GCT_receive( )` function blocks until a message is available, while the `GCT_grab( )` function returns immediately. Once the application has finished processing the received message, it should release the message structure back to the system by calling the `relm( )` library function. The library functions are described in the Software Environment Programmer's Manual.

To configure the board using individual messages, the following sequence should be used. The message sequence is shown diagrammatically in Figure 4. Protocol Configuration Message Sequence Diagram.

1. Build and send an SSD Reset Request (`SSD_MSG_RESET`) to the SSD module. This message contains the parameters required to initialize the SSD module.
2. Then build and send a Board Reset Request (`SSD_MSG_RST_BOARD`) for each board in the system. This message contains the address (or identifier) of the board and the name of the code file. It causes the board to be reset and the code file downloaded. For each board, the application should wait until a Board Status Indication (`SSD_MSG_STATE_IND`) is received and inspect the status field to determine if the reset operation was successful. On failure, the user should check carefully the `ssdm` parameters and try again.
3. Build and send a Board Configuration Request (`MGT_MSG_CONFIG0`) to the onboard management task (`MGMT_TASK_ID`) to configure the basic board parameters. When using Dialogic® DSI SS7MD Boards, the value of the `config_type` parameter in the Board Configuration Request must be set to 3. For this version of the message, the automatic configuration of MTP parameters is not supported. Wait for the confirmation message and check the status.
4. To set up the LIU and port for the T1/E1 ports, the LIU Configuration Request should be used. Wait for the confirmation message for each LIU and check the status.

For each link in the system:

5. Build and send a Layer 1 Configuration Request (`MGT_MSG_L1_CONFIG`) to set up the physical configuration parameters for the link. This message should be sent to the onboard management module. Wait for the confirmation message and check the status.

6. Build and send an MTP2 Link Configuration Request (SS7\_MSG\_CONFIG) to set up the MTP2 configuration parameters. See the MTP2 Programmer's Manual for the message definition. Wait for the confirmation message and check the status.
7. Build and send an MTP3 Module Reset Message (MTP\_MSG\_RESET) to reset the MTP3 module. See the MTP3 Programmer's Manual for the message definition. Wait for the confirmation message and check the status.
8. Build and send an MTP3 Module Configuration Request (MTP\_MSG\_CONFIG) to set up configuration parameters that relate to the MTP3 environment (number of link sets and links to support, module\_ids for user part modules etc.). See the MTP3 Programmer's Manual for the message definition. Wait for the confirmation message and check the status.

For each link set perform the following:

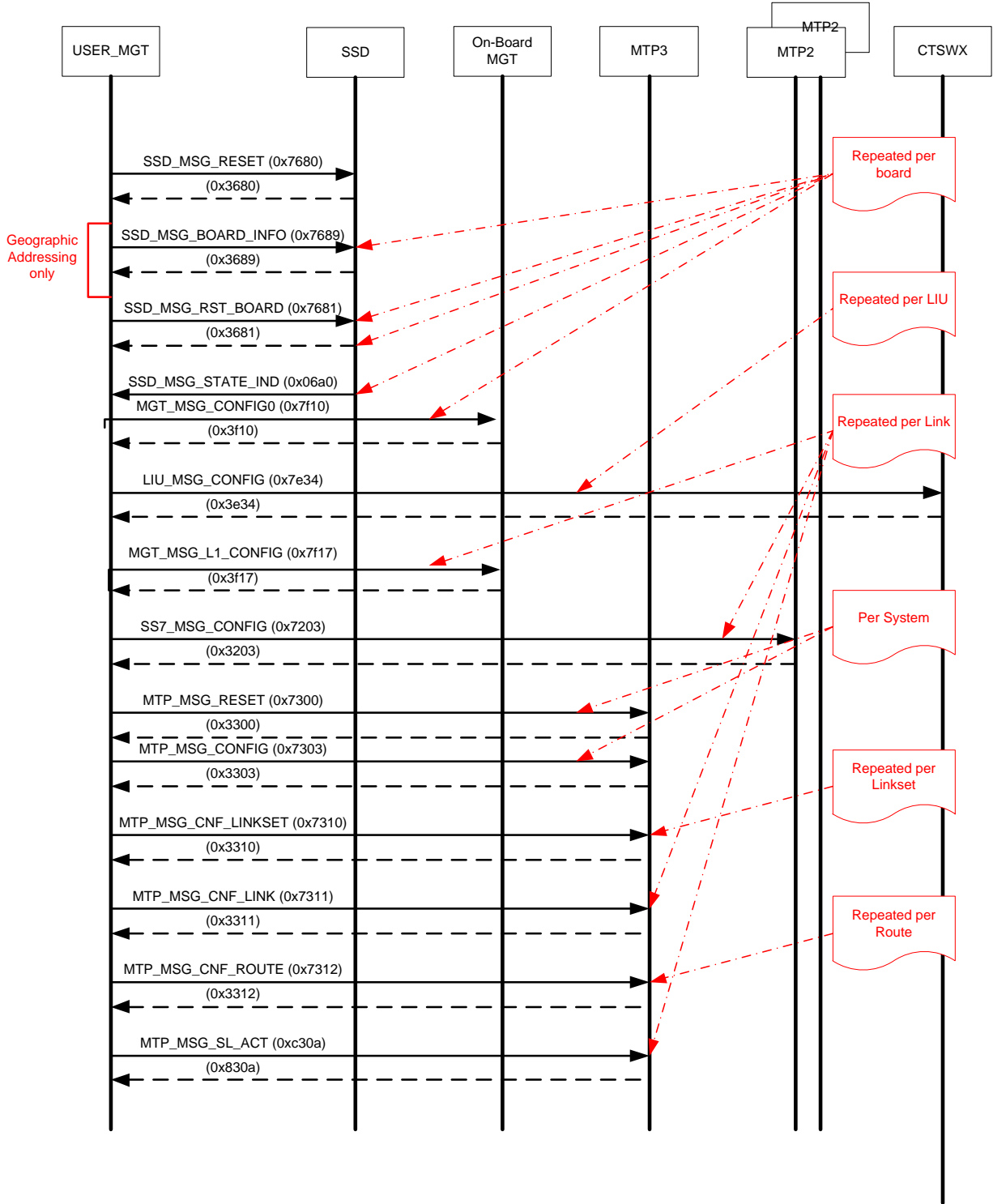
9. Build and send an MTP3 Link Set Configuration Request (MTP\_MSG\_CNF\_LINKSET) to set up configuration parameters for the individual link set (for example, local and adjacent point codes and the number of links in the link set). See the MTP3 Programmer's Manual for the message definition. Wait for the confirmation message and check the status.

For each link in the link set perform the following:

10. Build and send an MTP3 Signaling Link Configuration Request (MTP\_MSG\_CNF\_LINK) to set up configuration parameters for the individual link. See the MTP3 Programmer's Manual for the message definition. Wait for the confirmation message and check the status.
11. For each destination that needs to be accessed (including all adjacent signaling points), build and send an MTP Route Configuration Request (MTP\_MSG\_CNF\_ROUTE) to set up configuration parameters for the route. See the MTP3 Programmer's Manual for the message definition. Wait for the confirmation message and check the status.
12. Proceed now with the User Part configuration procedure. Once this is complete, issue an MTP Link Activation Request (MTP\_MSG\_ACT\_SL) for each link in the system as required to bring the link into service.

Further links, link sets and routes may be dynamically added at runtime using the same message sequences.

**Figure 4. Protocol Configuration Message Sequence Diagram**



## A.2 Monitoring Configuration Using Individual Messages

To configure the Dialogic® DSI SS7LD Network Interface Board for monitoring using individual messages, proceed as follows:

1. Build and send an SSD Reset Request to the SSD module. This contains the parameters to initialize the SSD module.
2. Build and send a Board Reset Request for each board in the system. This message contains the address (or identifier) of the board and the name of the code file. It causes the board to be reset and the code file downloaded. For each board, the application should wait until a Board Status Indication is received and inspect the status field to determine if the reset operation was successful. On failure, the user should check carefully the parameters and try again.
3. Build and send a Board Configuration Request (MGT\_MSG\_CONFIG0) to the onboard management task (MGMT\_TASK\_ID) to configure the basic board parameters. When using DSI SS7LD Boards, the value of the config\_type parameter in the Board Configuration Request must be set to 3. For this version of the message, the automatic configuration of MTP parameters is not supported. Wait for the confirmation message and check the status.
4. To set up the LIU and port for the T1/E1 ports, the LIU Configuration Request (LIU\_MSG\_CONFIG) should be used. For monitoring, the high\_Z parameter must be set to 2. Wait for the confirmation message for each LIU and check the status.

For each link in the system:

5. Build and send a Layer 1 Configuration Request (MGT\_MSG\_L1\_CONFIG) to set up the physical configuration parameters for the link. This message should be sent to the onboard management module. Wait for the confirmation message and check the status.
6. Build and send an MTP2 Link Configuration Request (SS7\_MSG\_CONFIG) to set up the MTP2 configuration parameters for monitoring operation. See the *Dialogic® SS7 Protocols MTP2 Programmer's Manual* for the message definition. Wait for the confirmation message and check the status.