



Creating Circuit-Switched Telephony Systems

Installation and Configuration

Executive Summary

This document describes the installation and configuration of circuit-switched telephony systems that combine Intel NetStructure® SS7 boards and Intel® Dialogic® media processing boards on platforms based on Linux* and Windows* operating systems.

Table of Contents

Introduction	1	Step 5 – Configuring the Global Call for SS7 Software 12	(gcss7.cfg)
System Description	1	Step 6 – Starting the Software	14
Hardware Requirements	2	Step 7 – Running the gc_basic_call_model Demo	15
Software Requirements	2	Troubleshooting	15
Step 1 – Installing the Intel® Dialogic® Software	2	Summary	15
Linux	2	Appendix A: Linux system.txt File	16
Windows	3	Appendix B: Linux config.txt File	17
Step 2 – Installing the SS7 Development Package		Appendix C: Linux gcss7.cfg.txt File	18
from Intel	6	Appendix D: Windows system.txt File	21
Linux	6	Appendix E: Windows config.txt File	22
Windows	6	Appendix F: Windows gcss7.cfg.txt File	23
Step 3 – Configuring the SS7 Software	6	Acronyms	25
system.txt	7	For More Information	25
config.txt	8		
Step 4 – Proving the SS7 Configuration	11		

Introduction

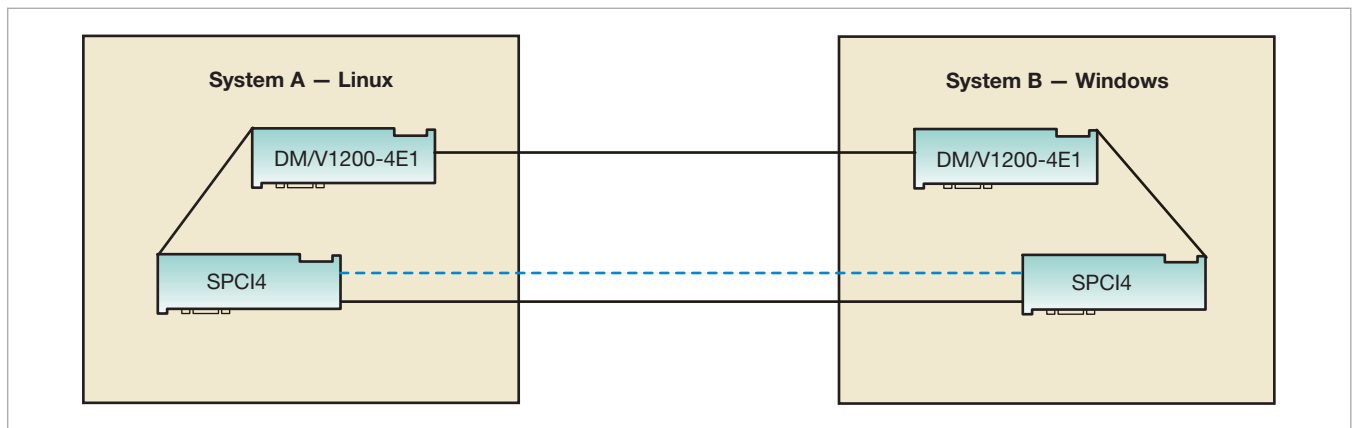
This application note guides the reader, step by step, through the installation, configuration, and run-time operation of two computer telephony systems that combine an Intel NetStructure® SS7 board and an Intel® Dialogic® media processing board.

These systems will be connected back-to-back using cross-over cables. The gc_basic_call_model demo will show how the Global Call library facilitates the development of SS7 telephony applications running on top of the ISDN User Part (ISUP) signaling protocol.

System Description

In this application note, System A runs Red Hat® Enterprise Linux® v3 and System B runs Microsoft® Windows XP®; however, readers are free to use the same operating system on both systems (see Figure 1). Since the configuration files given in the appendices are operating system agnostic, this should not cause any problems.

Figure 1. Sample System Configuration



Hardware Requirements

- 2 PCs with Pentium® III processors, 512 MB RAM
- 2 Intel NetStructure SPC14 SS7 boards with ISUP or ISUP-L license buttons
- 2 Intel NetStructure DM/V1200-4E1 voice boards (PCI form factor)
- 2 H.100 CT Bus cables
- 2 cross-over E-1 cables

Software Requirements

Linux Operating System

- Red Hat Enterprise Linux v3 Update 4
- The latest Intel NetStructure SS7 Development Package for Linux
- The latest SS7 code file for SPC14 (`ss7.dc3`)
- Intel Dialogic System Release 6.1 for Linux

Windows Operating System

- Microsoft Windows XP
- The latest Intel NetStructure SS7 Development Package for Windows
- The latest SS7 code file for SPC14 (`ss7.DC3`)
- Intel Dialogic System Release 6.0 PCI Service Update 30 (or greater) for Windows

Notes:

All SS7 software can be downloaded from <http://resource.intel.com/telecom/support/ss7/cd/hostprotocolsoftware/index.htm>

More information about Intel Dialogic System Releases can be found on <http://www.intel.com/network/csp/products/8317web.htm>

Step 1 – Installing the Intel® Dialogic® Software

Linux

Please refer to the System Release 6.1 for Linux Software Installation Guide (http://www.intel.com/design/network/manuals/telecom/sr611in/htmlfiles/release_install.htm) for detailed instructions on how to install the System Release software. Note that you only need to select “Intel NetStructure® DMV/DMN/DMT” and the Global Call SS7 Support Software packages at the installation menu.

Once the System Release has been successfully installed, you still need to start the `config.sh` script to copy and load the DM3 driver files. When this process completes, the DM/ board should be detected and you should be prompted with this menu.

```
Intel(R) Dialogic(R) Configuration Manager -
Main Screen
-----
1) Intel(R) NetStructure(TM) DM3 Board Summary
2) Intel(R) Dialogic(R) Board Summary (NO
   BOARDS)
3) Intel(R) NetStructure(TM) IPT Board Summary
4) TDM Bus Settings
-----
(s to save, x to save & quit, q to quit) the
configuration
? for help and ! for navigation help
You can only configure one board at a time.
Enter the number associated
with the product category of the board you want
to configure :1
```

Select item 1 to get the list of DM3 boards that were detected on your system:

```
Intel(R) NetStructure(TM) DM3 Board Summary
-----
You must configure or disable each board shown.
After a board is configured, a valid PCD file
name is displayed in the PCD File Name column.
-----
Thumb   Board           Log   PCD File
Wheel   Status   Model   ID   Name
-----
0)      C      DM/V1200-4E1  1   NOT_SET
-----
(s to save, x to save & quit, q to quit) the
configuration
p to return to Intel(R) Dialogic(R)
Configuration Manager - Main Screen
? for help and ! for navigation help
Enter the Thumb Wheel of the board to configure:
0
```

Then, enter the Thumb Wheel number of the board you want to configure (0 in this case).

```

Modify Board Settings
-----
These are the current settings for the board
selected:
Physical Slot..... : 0
Model Name..... : DM/V1200-4E1
Logical ID..... : 1
Board Status..... : Configured
PCD File Name..... : NOT_SET
CONFIG File Name... : NOT_SET
-----
The following items can be modified:
1) Specify the PCD File
2) Trunk Configuration
3) Protocol Development Kit (PDK) Configuration
   (NOT APPLICABLE)
4) Modify NIC Configuration (NOT APPLICABLE)
5) Copy Configuration From Board
6) Advanced Board Settings
-----
(s to save, x to save & quit, q to quit) the
configuration
p to return to Intel(R) NetStructure(TM) DM3
Board Summary
? for help and ! for navigation help
Enter the number of the item to modify: 1
    
```

Select item number 1 (Specify the PCD File) and you should be prompted with a list of available media loads for the board you are using.

```

Specify the PCD File
-----
A Product Configuration Description (PCD) file
must be selected to configure your board. The
corresponding CONFIG file will be automatically
selected.
The following PCD files are valid for your
board:
1) ml1_qs_dass2.pcd
2) ml1_qs_dpns.pcd
3) qs_isdn_net5.pcd
4) ml1_qs_net5.pcd
5) qs_isdn_qsigel.pcd
6) ml1_qs_qsigel.pcd
7) qs_r2mf.pcd
8) ml1_qs_r2mf.pcd
9) ml1_qs_ts16.pcd
-----
* is current setting
(s to save, x to save & quit, q to quit) the
configuration
p to return to Modify Board Settings
? for help and ! for navigation help
Enter the number corresponding to the PCD file
you want to select: 1
    
```

Please select a clear channel media load, such as:

- Item number 9 (ml1_qs_ts16.pcd) for a regular DM/V1200-4E1 board
- Item number x (ml2_qsa_ts16.pcd) for a DM/V1200A-4E1 board

From the Intel® Dialogic® Configuration Manager - Main Screen, type the number **4** and press **Enter** to select the TDM Bus Settings option. Then, select the "" menu (item number 2 in the TDM Bus Settings menu) and make sure that your DM/V board is set as "Secondary Master Board" (item number 2), since we want the SS7 board to be the primary CT Bus clock master.

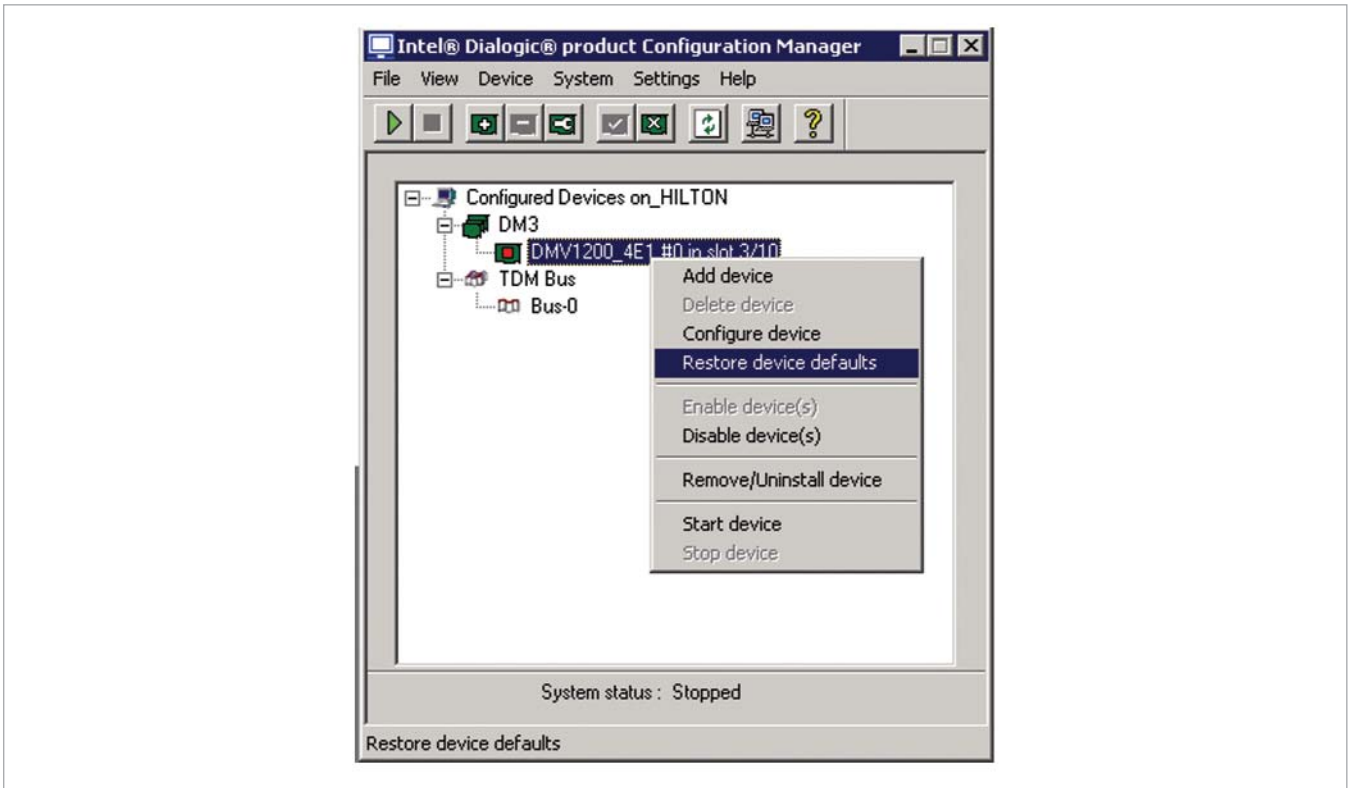
Windows

Please refer to the System Release 6.0 PCI for Windows Installation Guide (http://resource.intel.com/telecom/support/releases/winnt/sr60pci/onldoc/htmlfiles/release_install.html) for detailed installation instructions.

When the "Select Features" window appears during installation, please check these options:

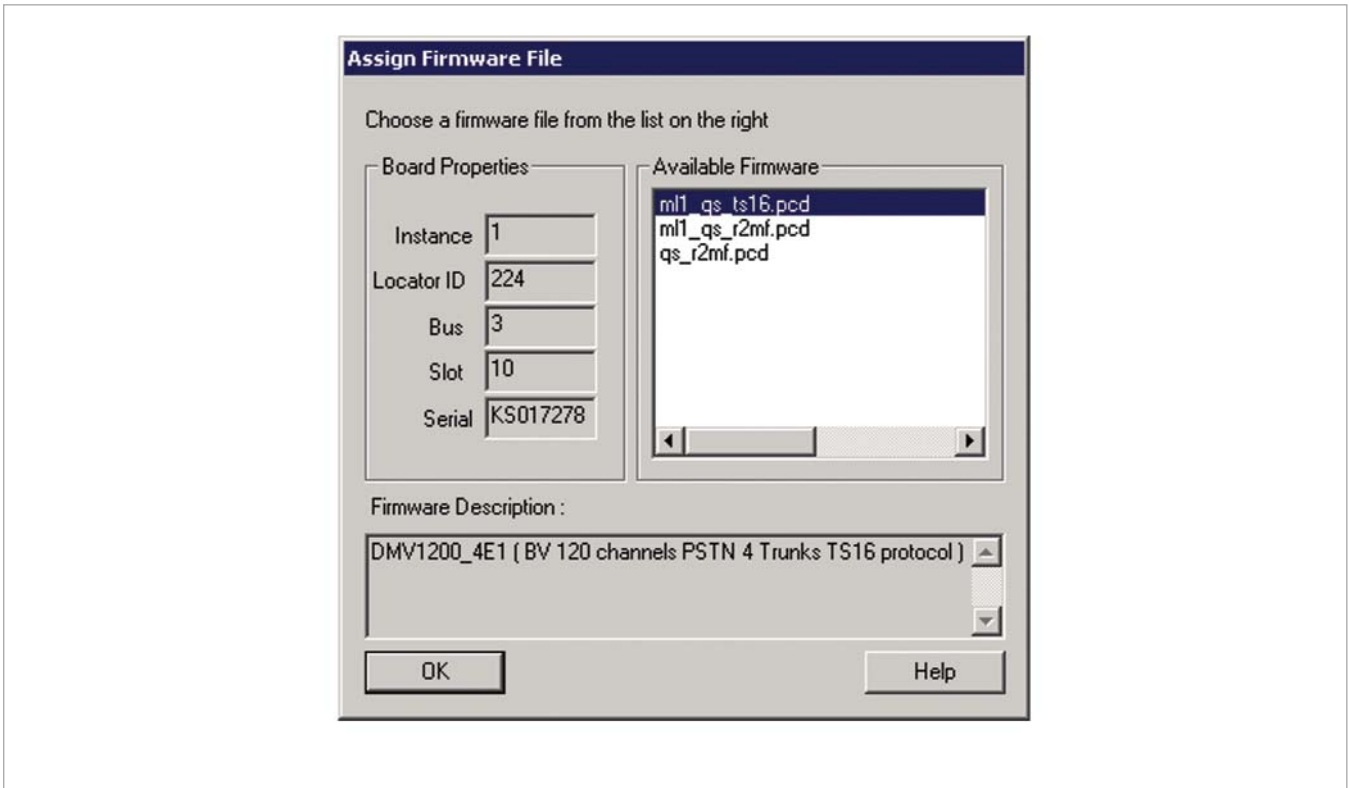
- **Development Package:** This allows for recompiling the application if this is needed.
- **Core Runtime Package:** This option installs all Intel telecom board drivers, firmware, and parameter files.
- **Demos:** This option installs all the sample programs for Intel telecom boards and, among others, the `gc_basic_call_model` application, which will be used to make and receive calls between the two systems.

Figure 2. Forcing a Hardware Detection



Once the System Release has been installed and the system has been rebooted, please open up the Dialogic Configuration Manager.

- Right click on the DM3 device and select “Restore Device Default” to force a re-detection of your Intel Dialogic hardware, as shown in Figure 2.
- Then pick up a clear channel media load (e.g., ml1_qs_ts16.pcd for a regular DM/V1200-4E1 board) as illustrated in Figure 3.

Figure 3. Selecting a Clear Channel Media Load

Note: If you can't find an appropriate clear channel media load for your DM3 board (e.g., a DM/V960-4T1 or an IP Link board), then manually edit the corresponding *.config file in order to configure the E-1 or T-1 trunk in clear channel mode. This can be achieved with the following procedure:

1. Identify the *.config file that corresponds to the media load you have chosen for your DM3 board (e.g., ml1_qs_cas.config) and open it in a text editor.
2. Search for the first occurrence of the 0x1602 parameter.

```
[lineAdmin.1]
...
SetParm=0x1602,4    ! SignalingType (CAS=4,
                    CCS=5, Clear=6)
...
```

3. Change the parameter value to 6 to configure the corresponding trunk in clear channel mode.

```
[lineAdmin.1]
...
SetParm=0x1602,6    ! SignalingType (CAS=4,
                    CCS=5, Clear=6)
...
```

4. Repeat this operation the other trunks you want to set in clear channel mode ([lineAdmin.2], [lineAdmin.3], etc.)
5. Generate a new *.fcd file using the fcdgen utility.

```
c : \Program Files \Dialogic \data \fcdgen
ml1_qs_cas
fcdgen Version 0.00      Build: 4
Copyright © Intel Corporation 2003
Building ml1_qs_cas.fcd from ml1_qs_cas.config
```

6. After having re-started the Intel Dialogic Service, you can verify that there are now 24 T-1 timeslots or 31 E-1 timeslots on the clear channel trunks, using "Devmapdump".

```
c : \Program Files \Dialogic \bin \Devmapdump >
a.txt
notepad a.txt
```

Step 2 – Installing the SS7 Development Package from Intel

First download the SS7 software components from the Intel Web site. This URL, <http://resource.intel.com/telecom/support/ss7/downloads/index.htm>, points to a page that offers different download sections.

Go to the Product Specific section and download the SS7 Binary (code file) for SPCI2S and SPCI4 signaling boards (Section 2.5). Keep the code file (**SS7.DC3**) in a directory on your system (usually the “Septel” directory for simplicity). On a Linux system, please rename **SS7.DC3** to **ss7.dc3**.

Then, go to the Generic Information section and, in Section 3.1, download the SS7 Development Package for Windows and the SS7 Development Package for Linux.

Linux

For Linux, the Intel SS7 Development Package comes in the form of a compressed **tar** file. You need to manually uncompress and extract the contents of this file (using the **tar** utility) into a directory you have previously created. Before attempting to load the SPCI driver module, you have to compile the object file from the driver source code using the **build_spici_cpm.sh** script file. This build script assumes a suitable environment for building Kernel modules is available. This must include the appropriate kernel include files found at: `/usr/src/linux-`uname -r`/include` (e.g., `/usr/src/linux-2.4.7-10/include`). If these are not found, the build will fail. Following that, the **install_spici_cpm.sh** script installs the device driver, automatically allocates a major device number, and creates the four appropriate device nodes.

Note: ‘**install_spici_cpm.sh**’ dynamically loads the driver module into the kernel. The module will remain loaded as long as the PC is not rebooted. After a system reboot, you will have to re-load the module using the **insmod** utility.

Windows

Installing the Intel SS7 Development Package for Windows is simpler than for Linux systems since there is an Install Shield script that copies the file and drivers to the appropriate directories. More info can be found in Section 2.1 of the SS7 Programmer’s Manual. However, the SPCI device driver must be manually registered to the system using the following command:

```
C:\> net start septel
```

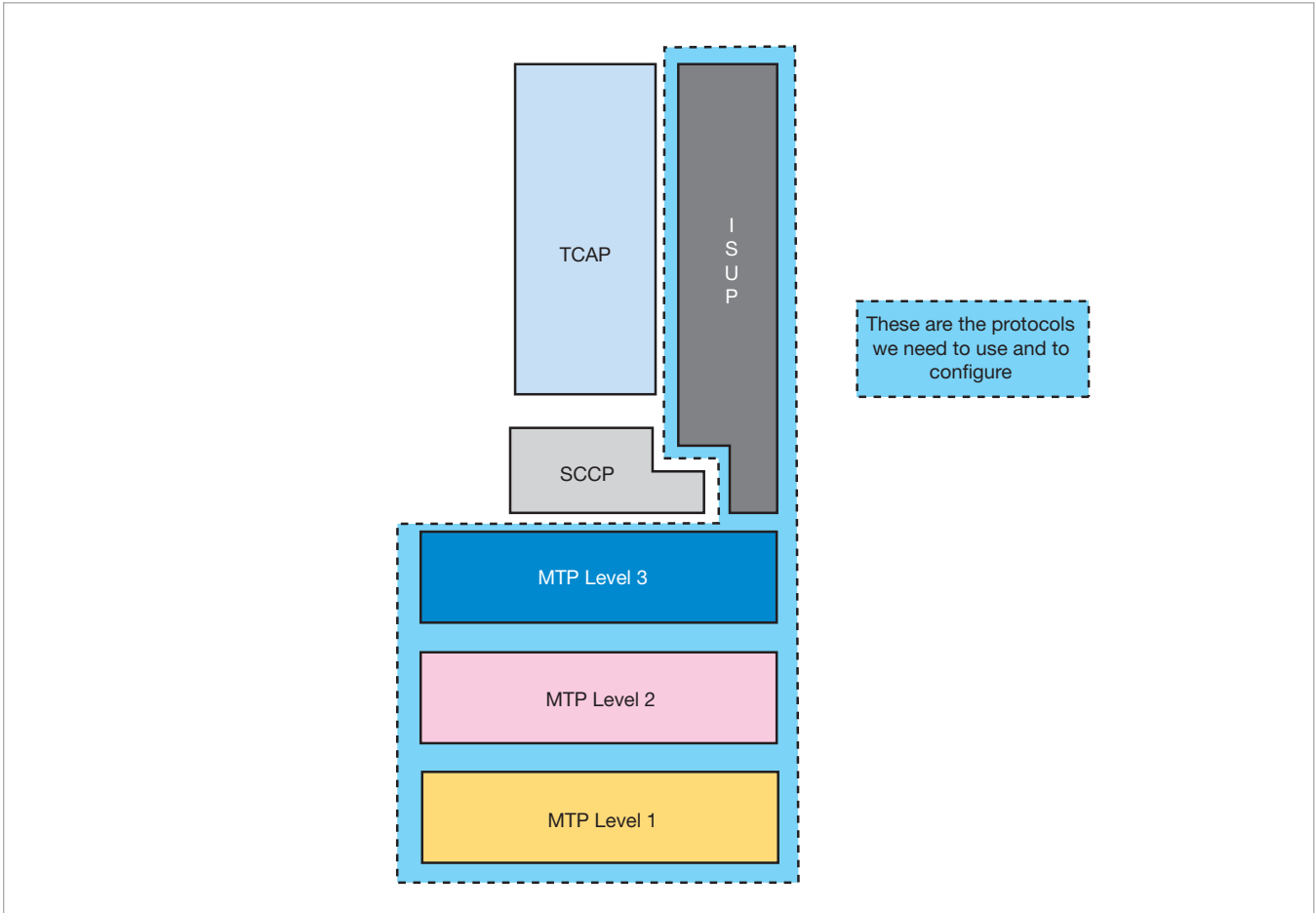
Once you have done this, open up the Windows Device Manager. Make sure to turn the Show Hidden devices option in the View menu. You should now be able to see the Septel device under the Non-Plug and Play Drivers. Double click on the Septel device, go to the Driver tab, and change the startup mode to automatic.

Step 3 – Configuring the SS7 Software (**config.txt** and **system.txt**)

Note: Before configuring the SS7 board and protocol stack, it is very important that readers get a clear understanding of the underlying SS7 concepts such as link, link set, point codes, routes etc. If you are unfamiliar with those concepts, we strongly recommend you look at the SS7 tutorial on the Intel Web site at <http://resource.intel.com/telecom/support/ss7/SS7tutorial/tutorial.html>. For more details about the SS7 architecture, we also recommend you read *ISDN & SS7 – Architectures for Digital Signaling Networks*.

There are two configuration files used when setting up the software for an Intel SS7 signaling board: **system.txt** and **config.txt**. This section will require some SS7 knowledge, specifically about the Message Transfer Part (MTP) layers and the ISDN User Part (ISUP). Figure 4 illustrates the protocols that need to be configured.

Figure 4. SS7 Protocol Stack



system.txt

The **system.txt** file specifies how Intel's SS7 software environment is set up. This software environment is commonly referred to as the **GCTLOAD** environment.

The easiest way to understand the operation of this SS7 software is to think of each protocol layer as an independent module. Each protocol module can run either on the host CPU or on the SS7 signaling board, except MTP1 and MTP2 which must always run on the SS7 board. Protocol modules use inter-process communication mechanisms to exchange messages with each other.

GCTLOAD modules are identified with a module ID. The module ID uniquely identifies protocol modules in the **GCTLOAD** environment. Any piece of software (e.g., the user application) that directly communicates with the SS7 protocol modules is also considered as a **GCTLOAD** module and hence must also be assigned its own module ID. You can learn more about Intel's SS7 **GCTLOAD** software environment in *Software Environment Programmer's Manual – Issue 4*.

Two working **system.txt** files (one for Linux, one for Windows) are included in Appendices A and D for this cross configuration. Please manually copy them in the directory where you have installed the Intel SS7 Development Package.

Let's do a quick review of the three major sections of the **system.txt** file.

LOCAL

The **LOCAL** commands assign module IDs for the modules that will run on the host CPU. The host machine is the computer where the application is running. Defining modules in this section also creates queues for incoming messages.

Our **system.txt** will include the following **LOCAL** commands:

- LOCAL 0x20** * Module ID for SSD
- LOCAL 0xcf** * Module ID for s7_mgt
- LOCAL 0xef** * Module ID for s7_log
- LOCAL 0x4d** * Module ID for the GC/SS7 service/daemon

Application Note Creating Circuit-Switched Telephony Systems

- **SSD (System Seven Dispatcher)** — This special host module is responsible for relaying messages from the host to the board
- **S7_mgt** — This configuration utility reads the `config.txt` file and configures the board and protocol modules (see `config.txt`; Appendix B for Linux, Appendix E for Windows)
- **S7_log** — General-purpose logger; very useful to analyze system run-time behavior and troubleshoot problems
- **GC/SS7 service daemon** — Server side of the Global Call for SS7 client/server architecture, responsible for receiving and sending SS7 messages from and to the ISUP protocol module

REDIRECT

The **REDIRECT** section redirects messages sent to one module ID to an alternative module. For example, if you **REDIRECT** messages sent to ISUP (0x23) to SSD (0x20), these messages will be sent to the board via the SSD process. This would be used where protocol modules are running on the board.

Since we are running MTP2, MTP3, and ISUP on the signaling board, we have to **REDIRECT** the messages for these layers.

```
REDIRECT 0x71 0x20 * MTP2 messages redirected to SS7 card
REDIRECT 0x22 0x20 * MTP2 messages redirected to SS7 card
REDIRECT 0x23 0x20 * ISUP messages redirected to SS7 card
```

FORK_PROCESS

The **FORK_PROCESS** section actually executes programs, such as protocol modules running on the host or management tools (e.g., `s7_mgt`). In our particular case, we don't have any protocols running on the host, so only management tools will be forked (e.g., `s7_log`, `s7_mgt`).

config.txt

The `config.txt` file is used for protocol configuration. It contains configuration commands for the MTP and ISUP protocols. `s7_mgt` reads the contents of this file and translates the configuration commands into management messages that are sent to the different protocol modules.

Two working `config.txt` files (one for Linux, one for Windows) are included in Appendices B and E for this cross configuration. Please manually copy them in the directory where you have installed the Intel SS7 Development Package.

For full description of all the protocol configuration commands for the SPCI4/SPCI2S product, please refer to Appendix A of the SS7 Programmer's Manual.

Physical Configuration

The `SEPTELPCI_BOARD` command defines the settings of the board itself. It allows defining the clock source for the board, and identifies what protocol modules are running on the board.

```
*          <board_id>    <flags>          <code_file>      <run_mode>
SEPTELPCI_BOARD  0      0x0043          ss7.dc3         ISUP
```

<flags> - Possible values

CTbus Slave	0x00C2
CTbus Master – clock derived from one of the SS7 card line interface	0x0043
CTbus Master – clock derived from SS7 card internal oscillator	0x0042

In this case, set `<flags>` on System A to 0x0042 and on System B to 0x0043, so that System B synchronizes on the clock generated by System A.

The `LIU_CONFIG` command configures the operating parameters for an E-1/T-1 trunk interface. These parameters include the line encoding, the frame format, and whether cyclical redundancy check (CRC) is enabled or not.

```
*          <board_id>   <liu_id>       <liu_type>     <line_code>    <frame_format> <crc_mode>
LIU_CONFIG 0           0           5             1             1             1
```

MTP Configuration

Let's clearly distinguish the three types of point codes we need to configure.

- **Local Point Code (LPC)** — This is the SS7 address of the signaling point we are configuring. Assign Point Code 1 to System A and Point Code 2 to System B.
- **Adjacent Point Code (APC)** — This is the SS7 address of the adjacent signaling point; this is the node the system has physical connection(s) to. For System A, the APC is Point Code 2. For System B, it is Point Code 1.
- **Destination Point Code (DPC)** — This is the SS7 address of the remote signaling point; this is the node the system has to exchange SS7 messages with. In this case (back-to-back configuration), the APC is also the DPC.

`MTP_CONFIG` is the main configuration command for the MTP module. The first two parameters are not used and can be left to 0. The `<options>` field allows configuring the variant being used (ITU or ANSI).

- For ITU operation, all bits can be set to 0.
- For ANSI operation, bits 8, 9, 10, and 11 must all be set to 1, leading to a value of 0x00000f00.

```
*          <reserved1>   <reserved2>   <options>
MTP_CONFIG 0           0             0x00000000
```

The `MTP_LINKSET` command is used for the configuration of a link set to an adjacent signaling point.

```
*          <linkset_id>   <adjacent_spc> <num_links>    <flags>       <local_spc>   <ssf>
MTP_LINKSET 0           2             1             0x0000       1             0x8
```

The `<ssf>` parameter is a 4-bit value made up (DCBA) with:

D C	Network type
0 0	International network
0 1	International spare
1 0	National network
1 1	National spare

B A	Priority value
0 0	0
0 1	1
1 0	2
1 1	3

In ITU, the most common SSF value for MTP3 messages is 0x8. In ANSI operation, the most common SSF value for MTP3 is 0xb.

The `MTP_LINK` command is used to define the links within every link set.

```
*          <link_id> <linkset_id> <link_ref> <slc> <board_id> <blink> <stream> <timeslot> <flags>
MTP_LINK 0       0       0       0       0       0       0       16       0x0006
```

Application Note Creating Circuit-Switched Telephony Systems

Notes:

<blink> refers to the signaling processor resource that will handle that particular SS7 link.

<stream> can refer to two different things.

- When the signaling link is on an E-1/T-1 trunk that is physically connected to the SS7 board, <stream> is the actual E-1/T-1 port ID (ranging from 0 to 3 for an SPCI4 board) that is used for that particular trunk. In this situation, the <timeslot> refers to the actual timeslot position on that trunk (e.g., timeslot 16).
- When the signaling link is on an E-1/T-1 trunk that is physically connected on another board in the chassis (such as a DM/N or DM/V board), <stream> must be set to 0x83. In this case, the <timeslot> refers to an internal bus slot in the SS7 board. Normally, you would start with time slot 1 for the first link and increment this value for each additional link coming from a separate board.

The `MTP_ROUTE` command defines the DPC for the system, and specifies which link set needs to be used to reach it. Up to two link sets can be associated to a single DPC.

```
*          <dpc>          <norm_ls>          <user_part_mask>          <flags>          <second_ls>
MTP_ROUTE 567            0                0x0020                0x0000            0
```

Note: The <user_part_mask> is a 16-bit value with bit n (n=0...15) set to allow the route to be used for messages with Service Indicator n. The service indicators are defined as follows:

User Part	Service Indicator	<user_part_mask>
SCCP	3	0x0008
TUP	4	0x0010
ISUP	5	0x0020

ISUP Configuration

The `ISUP_CONFIG` command defines the ISUP characteristics: point codes, CIC codes, etc. The point codes in this section must match those in the MTP Configuration section.

Here we basically define voice circuits, how many and where they are supposed to go.

```
*          <res1>          <res2>          <user_id>          <options>          <num_grps>          <num_ccts>
ISUP_CONFIG 0              0                0x4d              0x0475            2                  64
```

Note: For ANSI operation, the <options> field of `ISUP_CONFIG`, which is a bit field, must have bits 8 and 9 set in order to enable ANSI-compliant operations. These bits are 0 by default.

The `ISUP_CFG_CCTGRP` command defines the configuration parameters for a group of ISUP circuits. Usually a circuit group is made of all the voice circuits on a single E-1 or T-1 trunk.

Note: About the <cic_mask> parameter, a circuit group may contain gaps in the CIC allocation. This is very usually the case in Europe, where SS7 service providers usually skip every multiple of 16. We will use this convention in our system set-up.

Application Note Creating Circuit-Switched Telephony Systems

If the physical connection is good, then you can try to bring the MTP link into service. To do so, please open another command window, go the SS7 directory, and type

```
./mtps1 act 0 0 (Linux)    or    mtps1 act 0 0 (Windows)
```

You should then get the following status indications in the `gctlload` window (Windows side):

```
S7L:I0000 Level 2 State : id=0 INITIAL ALIGNMENT
S7L:I0000 Level 2 State : id=0 ALIGNED READY
S7L:I0000 Level 2 State : id=0 IN SERVICE
S7L:I0000 MTP Event : linkset_id/link_ref=0000 Changeback
S7L:I0000 MTP Event : linkset_id=00 Link set recovered
S7L:I0000 MTP Event : linkset_id=00 Adjacent SP accessible
S7L:I0000 MTP Event : point code=00000001
Destination available
```

On the Linux side, you should get similar messages, with the only difference that the remote point code is 00000002.

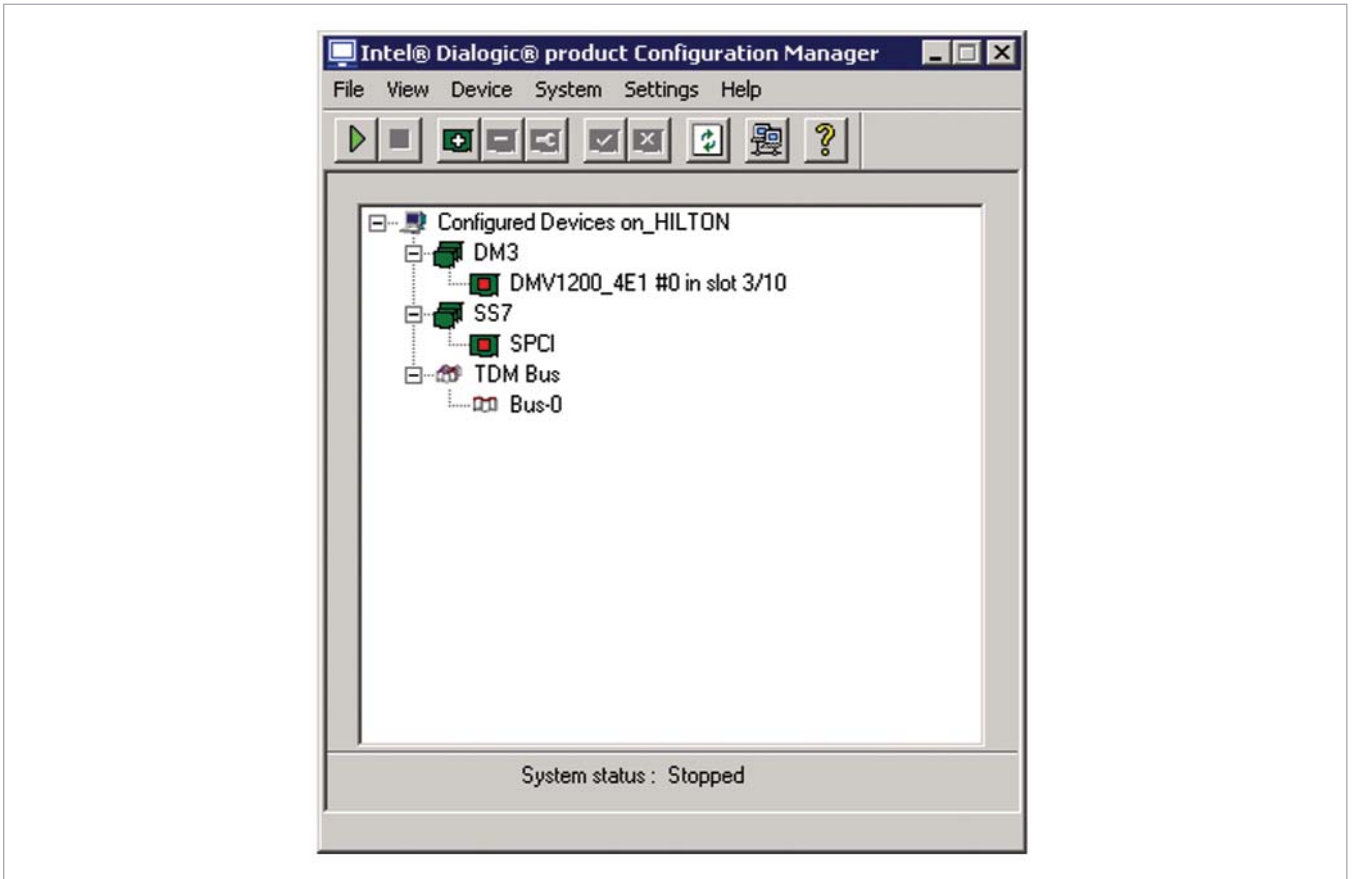
Step 5 – Configuring the Global Call for SS7 Software (`gcss7.cfg`)

The Intel Dialogic System Release contains a file called `gcss7.cfg`, which is used to configure the Global Call for SS7 software.

- On Linux systems, this file is located in `/usr/dialogic/cfg`.
- On Windows systems, this file is located in `$DIALOGICDIR\Dialogic\cfg` (e.g., `c:\Program Files\Dialogic\cfg`)

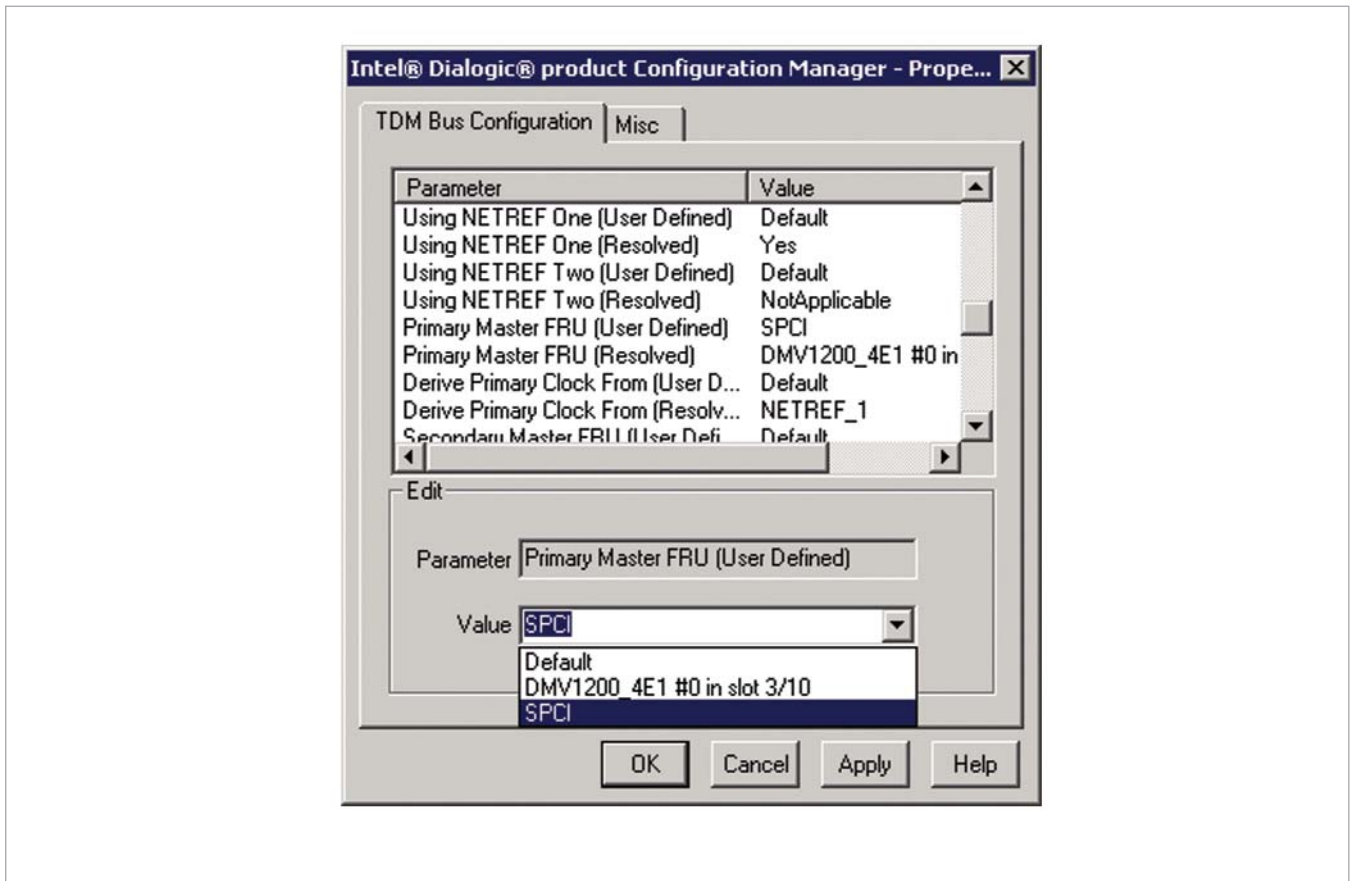
Working `gcss7.cfg` files are given in Appendices C and F. Please refer to Section 3.3 of the *Global Call for SS7 Technology Guide* for a detailed description of the `gcss7.cfg` configuration commands and parameters.

Once the `gcss7.cfg` has been properly updated, Windows users may close the DCM and re-open it. Then, the SPCI board should appear in the device tree, as shown in Figure 5.

Figure 5. DCM Showing SS7 Board after `gcss7.cf` Modifications

At this time, it is very important to double-click on the 'Bus-0' device, in order to configure the SPCI board as the CT Bus Primary Master. This is illustrated in Figure 6.

Figure 6. Configuring the SPCI Board as the TDM Bus Primary Master



Step 6 – Starting the Software

Once everything has been configured, starting the software requires two steps.

1. Start-up the Intel SS7 environment, using `gctload (-d)`. You must keep the `gctload` program running at all times.
2. Start-up the Intel Dialogic environment.
 - Linux: Invoke the `dlstart` script in `/usr/dialogic/bin`.
 - Windows: Click on the green triangle in the DCM.

Since the Global Call for SS7 software has been enabled to automatically activate the SS7 links, there is no need to manually do this with `mtps1`. If everything goes well, the Intel Dialogic environment will start without any errors and the SS7 links will be activated, resulting in the following indication messages to be printed in the `gctload` window:

```
S7L:I0000 Level 2 State : id=0 INITIAL ALIGNMENT
S7L:I0000 Level 2 State : id=0 ALIGNED READY
S7L:I0000 Level 2 State : id=0 IN SERVICE
S7L:I0000 MTP Event : linkset_id/link_ref=0000 Changeback
S7L:I0000 MTP Event : linkset_id=00 Link set recovered
S7L:I0000 MTP Event : linkset_id=00 Adjacent SP accessible
S7L:I0000 MTP Event : point code=00000001
```

Destination available

Step 7 – Running the gc_basic_call_model Demo

Edit the `gc_basic_call_model.cfg` file, un-comment the SS7 device, and rename it to `dkB1T1`.

- Linux system: Set the Direction to “Out”.
- Windows system: Set the Direction to “In”.

Run the application binary on each system, you should see `GCEV_UNBLOCKED` on each side and, after a short period of time, calls will start to be sent from the Linux system to the Windows system. Use **CTRL-C** to quit the demo.

Troubleshooting

SS7 related problems can be diagnosed in different ways:

If your problem is related to Intel’s SS7 hardware or software configuration, add the “debug mode” option to `ssds` and `s7_mgt` in your `system.txt` file:

```
FORK_PROCESS ssds.exe -d
FORK_PROCESS s7_mgt.exe -d
```

When starting up the `gctload` environment, use the `-d` option as well.

If your problem is related to the integration of Intel’s SS7 products in the Intel Dialogic System Release, then you need

to enable Global Call for SS7 service traces (via `gcss7.cfg`). This will result in a binary file called `D1gcS7.log`. You need to use a decoder tool, called `ss7trace`, to convert this binary log file into an ASCII file. Please refer to Section 7.2 of the *Global Call for SS7 Technology Guide* for more details on Global Call for SS7 service traces.

Note that it is possible to collect both `s7_log` and Global Call for SS7 service traces in a single file. To do this, you have to redirect the `s7_log` module (0xef) to the Global Call for SS7 service module (0x4d) in your `system.txt` file:

```
REDIRECT 0xef 0x4d
```

If your problem is related to the Global Call for SS7 application, you can also enable Global Call for SS7 library traces in `gcss7.cfg`. This will result in a binary file called `ss7.log`. You need to use a decoder tool, called `ss7trace`, to convert this binary log file into an ASCII file. Please refer to Section 7.1 of the *Global Call for SS7 Technology Guide* for more details on Global Call for SS7 library traces.

Summary

This application note has described all the steps needed to set up a pair of telephony systems that combine Intel Dialogic media processing boards and Intel NetStructure SS7 interface cards.

Appendix A: Linux system.txt File

```

*****
*
* Linux system.txt file
*
*****
*
* Essential modules running on host:
*
LOCAL          0x20          * ssd/ssds/ssdh - Board interface task
LOCAL          0x00          * tim_xxx - Timer task
*
* Optional modules running on the host:
*
LOCAL          0xcf          * s7_mgt - Management/config task
LOCAL          0xef          * s7_log - Display and logging utility
LOCAL          0x4d          * GC/SS7 daemon
*
* Modules that optionally run on the host:
*
*LOCAL         0x23          * ISUP module
*LOCAL         0x4a          * TUP module
*LOCAL         0x33          * SCCP module
*LOCAL         0x14          * TCAP module
*LOCAL         0x22          * MTP3 module
*
* Essential modules running on the board (all redirected via ssd):
*
REDIRECT       0x71          0x20          * MTP2 module (except SS7HD boards)
* REDIRECT     0x81          0x20          * MTP2 module_id for SP 0 (SS7HD boards only)
* REDIRECT     0x91          0x20          * MTP2 module_id for SP 1 (SS7HD boards only)
* REDIRECT     0xe1          0x20          * MTP2 module_id for SP 2 (SS7HD boards only)
* REDIRECT     0xf1          0x20          * MTP2 module_id for SP 3 (SS7HD boards only)
REDIRECT       0x10          0x20          * CT bus/Clocking control module
REDIRECT       0x8e          0x20          * On-board management module
*
* Modules that optionally run on the board (all redirected via ssd):
*
REDIRECT       0x23          0x20          * ISUP module
* REDIRECT     0x4a          0x20          * TUP module
* REDIRECT     0x33          0x20          * SCCP module
* REDIRECT     0x14          0x20          * TCAP module
REDIRECT       0x22          0x20          * MTP3 module
*
* Redirection of status indications:
*
REDIRECT       0xdf          0xef          * LIU/MTP2 status messages -> s7_log
*
* Now start-up all local tasks:
* (For PCCS6 start-up use ssd,
* for SPCI4/SPCI2S/CPM8 start-up use ssds and
* for SS7HD boards use ssdh)
*
* FORK_PROCESS ./ssd
FORK_PROCESS  ./ssds
* FORK_PROCESS ./ssdh
FORK_PROCESS  ./tim_lnx
FORK_PROCESS  ./tick_lnx
FORK_PROCESS  ./s7_mgt
FORK_PROCESS  ./s7_log
*
*****

```

Appendix B: Linux config.txt File

```

*****
*
* Linux config.txt
*
*****
*
*
* Configure individual boards:
* For SPIC4 / SPIC2S boards:
* SEPTELPCI_BOARD <board_id> <flags> <code_file> <run_mode>
SEPTELPCI_BOARD 0 0x0042 ss7.dc3 ISUP
*
*
* Configure individual E1/T1 interfaces:
* LIU_CONFIG <board_id> <liu_id> <liu_type> <line_code> <frame_format>
*           <crc_mode>
LIU_CONFIG 0 0 5 1 1 1
*
*
* MTP parameters:
*
* MTP_CONFIG <reserved> <reserved> <options>
MTP_CONFIG 0 0 0x00000000
*
* Define linksets:
* MTP_LINKSET <linkset_id> <adjacent_spc> <num_links> <flags> <local_spc> <ssf>
MTP_LINKSET 0 2 2 0x0000 1 0x08
*
* Define signaling links:
* MTP_LINK <link_id> <linkset_id> <link_ref> <slc> <board_id> <blink>
*           <stream> <timeslot> <flags>
* For SPIC4 / SPIC2S / CPM8 boards:
MTP_LINK 0 0 0 0 0 0 0 0 16 0x0006
*
* Define a route for each remote signaling point:
* MTP_ROUTE <dpc> <linkset_id> <user_part_mask>
MTP_ROUTE 2 0 0x0020
*
*
* ISUP parameters:
*
* Configure ISUP module:
* ISUP_CONFIG <reserved> <reserved> <user_id> <options> <num_grps> <num_cts>
ISUP_CONFIG 0 0 0x4d 0x0475 2 64
*
* Configure ISUP circuit groups:
* ISUP_CFG_CCTGRP <gid> <dpc> <base_cic> <base_cid> <cic_mask> <options>
*           <user_inst> <user_id> <opc> <ssf> <variant> <options2>
ISUP_CFG_CCTGRP 0 2 0x01 0x01 0x7fff7fff 0x001c 0 0x4d 1 0x8 0 0x00
ISUP_CFG_CCTGRP 1 2 0x21 0x21 0x7fff7fff 0x001c 0 0x4d 1 0x8 0 0x00
*
*****

```

Appendix C: Linux gcss7.cfg File

```
#
# Intel (Dialogic) GlobalCall SS7 Configuration File (gcss7.cfg)
#
#   for Linux
#

#####
# Type of System Configuration #
#####

# Leave commented out or set to "None" when not using Dialogic SS7.
# Depending on the value of this parameter, the sections below, that
# are specific to some configurations (SeptelCard, SIU, SIU.Dual) will be
# used or not.
# Format: String - ["None", "Card", "SIU", "DualSIU"]
System.Configuration = "Card"

#####
# Parameters for the GlobalCall SS7 Call Control Library #
#####

# If defined, this parameter will cause the library logging to be
# activated at the first gc_Open() of an SS7 circuit and the trace
# file will have the specified name.
# Format: String
Library.LogFile = "ss7.log"

# Logging Level for the library
# Format: String - ["None", "Errors", "All"]
# Default: "Errors" (and Warnings)
Library.LogLevels = "All"

# Maximum size of the library log in kilobytes
# Format: Integer, Default: 200
#Library.LogMaxSize = 200

#####
# Parameters for the Dialogic SS7 service/daemon #
#####

# Logging Level for the service (Dlgs7.log)
# Format: String - ["None", "Errors", "All"]
# Default: "Errors" (and Warnings)
Service.LogLevels = "All"

# Maximum size of the service log in kilobytes
# Format: Integer, Default: 200
#Service.LogMaxSize = 200

# Does the service need to start GCTLOAD automatically?
# Format: String - ["Yes", "No"]
Service.GCTLOAD_Control = "No"

# Path to GCTLOAD (Used only if GCTLOAD_Control is set to 'Yes')
# For Setpel Cards, the parameter defaults to the same path as ConfigDir
# Format: String
#Service.GCTLOAD_Path = "/usr/septel"
```

```

# GCT-environment module id used by the service
# Format: Integer, Default: 0x4d
#Service.ModuleID = 0x4d

# Maximum timeout (in seconds) for server-application keep-alive mechanism
# Format: Integer; Default: 7; 0 means the mechanism is off
#Service.WatchDogMaxTime = 7

# Time (in ms) during which to accumulate Circuit Group Supervision Requests
# (reset, block, unblock) for a circuit group.
# Format: Integer - Default: 500
#Service.GroupCommandTimer = 500

#####
# Configuration for Septel Card Systems #
#####

# Path to the config.txt file
# Format: String
SeptelCard.ConfigDir = "/usr/septel"

# Should MTP links be automatically activated ?
# Format: String - ["None", "All"]
SeptelCard.Auto_Links_Activation = "All"

#####
# Configuration for SIU Systems #
#####

# ID of this host - Use 0 if only one host accessing the SIU(s)
# Format: Integer
SIU.HostID = 0

# SIU A - IP Address
# Format: String
#SIU.A.IP_Address = "111.112.113.114"

# SIU A - Account to use to connect to SIU when using FTP
# Format: String
#SIU.A.FTP_Account = "ftp"

# SIU A - Password for the FTP account
# Format: String
#SIU.A.FTP_Password = "ftp"

# SIU A - Directory to which to change (in FTP session) in order to get config.txt
# Format: String
#SIU.A.RemoteConfigDir = "."

# Maximum time (in seconds) to wait at startup for an SIU to come on-line before
# considering it as being down.
# Format: Integer, Default: 10
#SIU.InitTimeout = 10

#####
# Parameters specific to Dual-Resilient SIU Configurations #
#####

# SIU B Parameters - See the same parameters for SIU.A

```

Application Note Creating Circuit-Switched Telephony Systems

```
#SIU.B.IP_Address = "111.112.113.114"

#SIU.B.FTP_Account = "ftp"

#SIU.B.FTP_Password = "ftp"

#SIU.B.RemoteConfigDir = "."

# Max time (in seconds) to wait for group (de)activation command
# responses from SIU.
# Format: Integer, Default: 5
#SIU.Dual.SiuCommandTimeout = 5

# Debounce time (in seconds) for SIU Down indications
# Format: Integer, Default: 8
#SIU.Dual.SiuUpDebounceTime = 8

# Maximum number of retries for SIU group (de)activation commands
# Format: Integer, Default: 5
#SIU.Dual.MaxCmdRetries = 5

#####
# Parameters that are related to config.txt #
#####

# MTP Link source, link ID must match the value in config.txt.
# MtpLink <link_id> <"link_source">

# Circuit Group configuration, Group ID must match the value in config.txt.
# CGrp <gid> <"trunk_name"> [<base_TS> [<"Pref_SIU">]]
CGrp 0 dkB1
CGrp 1 dtiB1
#
# End of gcss7.cfg
#
```

Appendix D: Windows system.txt File

```

*****
*
* Windows system.txt
*
*****
*
* Essential modules running on host:
*
LOCAL          0x20          * ssd/ssds/ssdh - Board interface task
LOCAL          0x00          * tim_nt - Timer task
*
* Optional modules running on the host:
*
LOCAL          0xcf          * s7_mgt - Management/config task
LOCAL          0xef          * s7_log - Display and logging utility
LOCAL          0x4d          * GC/SS7 service
*
* Modules that optionally run on the host
*
*LOCAL         0x23          * ISUP module
*LOCAL         0x4a          * TUP module
*LOCAL         0x33          * SCCP module
*LOCAL         0x14          * TCAP module
*LOCAL         0x22          * MTP3 module
*
* Essential modules running on the board (all redirected via ssd):
*
REDIRECT       0x71  0x20    * MTP2 module (except SS7HD boards)
* REDIRECT     0x81  0x20    * MTP2 module_id for SP 0 (SS7HD boards only)
* REDIRECT     0x91  0x20    * MTP2 module_id for SP 1 (SS7HD boards only)
* REDIRECT     0xe1  0x20    * MTP2 module_id for SP 2 (SS7HD boards only)
* REDIRECT     0xf1  0x20    * MTP2 module_id for SP 3 (SS7HD boards only)
REDIRECT       0x10  0x20    * CT bus/Clocking control module
REDIRECT       0x8e  0x20    * On-board management module
*
* Modules that optionally run on the board (all redirected via ssd):
*
REDIRECT       0x23  0x20    * ISUP module
* REDIRECT     0x4a  0x20    * TUP module
* REDIRECT     0x33  0x20    * SCCP module
* REDIRECT     0x14  0x20    * TCAP module
REDIRECT       0x22  0x20    * MTP3 module
*
* Redirection of status indications:
*
REDIRECT       0xdf  0xef    * LIU/MTP2 status messages -> s7_log
*
* Now start-up all local tasks:
* (For PCCS6 start-up use ssd.exe and ssd_poll.exe,
* for SPCI4/SPCI2S/CPM8 start-up use ssds.exe and
* for SS7HD boards use ssdh.exe)
*
* FORK_PROCESS ssd.exe
* FORK_PROCESS ssd_poll.exe
FORK_PROCESS  ssds.exe
* FORK_PROCESS ssdh.exe
FORK_PROCESS  tim_nt.exe
FORK_PROCESS  tick_nt.exe
FORK_PROCESS  s7_mgt.exe
FORK_PROCESS  s7_log.exe
*
*****

```

Appendix E: Windows config.txt File

```

*****
*
* Windows config.txt
*
*****
*
* Configure individual boards:
* For SPCI4 / SPCI2S boards:
* SEPTELPCI_BOARD <board_id> <flags> <code_file> <run_mode>
SEPTELPCI_BOARD 0 0x0043 ss7.dc3 ISUP
*
*
* Configure individual E1/T1 interfaces:
* LIU_CONFIG <board_id> <liu_id> <liu_type> <line_code> <frame_format>
* <crc_mode>
LIU_CONFIG 0 0 5 1 1 1
*
*
* MTP parameters:
*
* MTP_CONFIG <reserved> <reserved> <options>
MTP_CONFIG 0 0 0x00000000
*
* Define linksets:
* MTP_LINKSET <linkset_id> <adjacent_spc> <num_links> <flags> <local_spc> <ssf>
MTP_LINKSET 0 1 2 0x0000 2 0x08
*
* Define signaling links:
* MTP_LINK <link_id> <linkset_id> <link_ref> <slc> <board_id> <blink>
* <stream> <timeslot> <flags>
* For SPCI4 / SPCI2S / CPM8 boards:
MTP_LINK 0 0 0 0 0 0 0 16 0x0006
*
* Define a route for each remote signaling point:
* MTP_ROUTE <dpc> <linkset_id> <user_part_mask>
MTP_ROUTE 1 0 0x0020
*
* Define any user provided Layer 4 protocol:
* MTP_USER_PART <service_ind> <module_id>
*MTP_USER_PART 0x0a 0x2d
*
*
* ISUP parameters:
*
* Configure ISUP module:
* ISUP_CONFIG <reserved> <reserved> <user_id> <options> <num_grps> <num_ccts>
ISUP_CONFIG 0 0 0x4d 0x0475 2 64
*
* Configure ISUP circuit groups:
* ISUP_CFG_CCTGRP <gid> <dpc> <base_cic> <base_cid> <cic_mask> <options>
* <user_inst> <user_id> <opc> <ssf> <variant> <options2>
ISUP_CFG_CCTGRP 0 1 0x01 0x01 0x7fff7fff 0x001c 0 0x4d 2 0x8 0 0x00
ISUP_CFG_CCTGRP 1 1 0x21 0x21 0x7fff7fff 0x001c 0 0x4d 2 0x8 0 0x00
*
*****

```

Appendix F: Windows gcss7.cfg File

```

#
# Intel (Dialogic) GlobalCall SS7 Configuration File (gcss7.cfg)
#
#   for Windows
#

#####
# Type of System Configuration #
#####

# Leave commented out or set to "None" when not using Dialogic SS7.
# Depending on the value of this parameter, the sections below, that
# are specific to some configurations (SeptelCard, SIU, SIU.Dual) will be
# used or not.
# Format: String - ["None", "Card", "SIU", "DualSIU"]
System.Configuration = "Card"

#####
# Parameters for the GlobalCall SS7 Call Control Library #
#####

# If defined, this parameter will cause the library logging to be
# activated at the first gc_Open() of an SS7 circuit and the trace
# file will have the specified name.
# Format: String
Library.LogFile = "ss7.log"

# Logging Level for the library
# Format: String - ["None", "Errors", "All"]
# Default: "Errors" (and Warnings)
Library.LogLevels = "All"

# Maximum size of the library log in kilobytes
# Format: Integer, Default: 200
#Library.LogMaxSize = 200

#####
# Parameters for the Dialogic SS7 service/daemon #
#####

# Logging Level for the service (Dlgs7.log)
# Format: String - ["None", "Errors", "All"]
# Default: "Errors" (and Warnings)
Service.LogLevels = "All"

# Maximum size of the service log in kilobytes
# Format: Integer, Default: 200
#Service.LogMaxSize = 200

# Does the service need to start GCTLOAD automatically?
# Format: String - ["Yes", "No"]
Service.GCTLOAD_Control = "No"

# Path to GCTLOAD (Used only if GCTLOAD_Control is set to "Yes")
# For Setpel Cards, the parameter defaults to the same path as ConfigDir
# Format: String
Service.GCTLOAD_Path = "c:\Septel"

# GCT-environment module id used by the service

```

Application Note Creating Circuit-Switched Telephony Systems

```
# Format: Integer, Default: 0x4d
Service.ModuleID = 0x4d

# Maximum timeout (in seconds) for server-application keep-alive mechanism
# Format: Integer; Default: 7; 0 means the mechanism is off
Service.WatchDogMaxTime = 0

#####
# Configuration for Septel Card Systems #
#####

# Path to the config.txt file
# Format: String
SeptelCard.ConfigDir = "c:\Septel"

# Should MTP links be activated automatically?
# Format: String - ["None", "All"]
SeptelCard.Auto_Links_Activation = "All"

#####
# Configuration for SIU Systems #
#####

# ID of this host - Use 0 if only one host accessing the SIU(s)
# Format: Integer
#SIU.HostID = 0

# SIU A - IP Address
# Format: String
#SIU.A.IP_Address = "172.22.197.103"

# SIU A - Account to use to connect to SIU when using FTP
# Format: String
#SIU.A.FTP_Account = "ftp"

# SIU A - Password for the FTP account
# Format: String
#SIU.A.FTP_Password = ""

# SIU A - Directory to which to change (in FTP session) in order to get config.txt
# Format: String
#SIU.A.RemoteConfigDir = "."

#####
# Parameters specific to Dual-Resilient SIU Configurations #
#####

# SIU B Parameters - See the same parameters for SIU.A
#SIU.B.IP_Address = "111.112.113.114"

#SIU.B.FTP_Account = "ftp"

#SIU.B.FTP_Password = "ftp"

#SIU.B.RemoteConfigDir = "."

#####
# Parameters that are related to config.txt #
#####
```

```
# MTP Link source, link ID must match the value in config.txt.
# MtpLink <link_id> <"link_source">

# Circuit Group configuration, Group ID must match the value in config.txt.
# CGrp <gid> <"trunk_name"> [<base_TS> [<"Pref_SIU">]]
CGrp 0 dkB1
CGrp 1 dtiB1

#
# End of gcss7.cfg
#
```

Acronyms

APC	Adjacent point code
CIC	Circuit Identification Code
CRC	Cyclical redundancy check
DPC	Destination point code
ISUP	ISDN user part
LPC	Local point code
MTP	Message transfer part

For More Information

SS7 Programmer's Manual for SPC12S, SPC14 & CPM8 – Issue 2

<http://resource.intel.com/telecom/support/ss7/cd/productspecific/index.htm>

ISDN & SS7 – Architectures for Digital Signaling Networks, Uyles Black, 1997, Prentice Hall ISBN 0-13-259193-6

Software Environment Programmer's Manual – Issue 4

<http://resource.intel.com/telecom/support/ss7/cd/genericinfo/index.htm>

Global Call for SS7 Technology Guide – March 2005

http://resource.intel.com/telecom/support/releases/linux/sr61lin/Onldoc/htmlfiles/globalcall_for_ss7.htm

To learn more, visit us at <http://www.intel.com>.

1515 Route Ten
Parsippany, NJ 07054
Phone: 1-973-993-3000

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

Intel products are not intended for use in medical, life saving, life sustaining, critical control or safety systems, or in nuclear facility applications.

Intel may make changes to specifications, product descriptions, and plans at any time, without notice.

Intel, Intel Dialogic, Intel NetStructure, Pentium, and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

*Other names and brands may be claimed as the property of others.

