CT Bus Clock Fallback for Linux Operating Systems

Demo Guide

August 2005

05-1900-002

Publication Date: August 2005

Document Number: 05-1900-002

Intel Converged Communications, Inc.
1515 Route 10
Parsippany, NJ 07054

For **Technical Support**, visit the Intel Telecom Support Resources website at:
http://developer.intel.com/design/telecom/support

For **Products and Services Information**, visit the Intel Telecom Products website at:
http://www.intel.com/design/network/products/telecom

For **Sales Offices** and other contact information, visit the Where to Buy Intel Telecom Products page at:
http://www.intel.com/buy/networking/telecom.htm

Contents

Tables

Revision History

This revision history summarizes the changes made in each published version of this document.

Document No.	Publication Date	Description of Revisions
05-1900-002	August 2005	Primary Master Fallback section: Modified the directory path for the tblist utility. Files Used by the Demo section: Added demo files to table in this section.
05-1900-001	September 2002	Initial version of document.

About This Publication

The following topics provide information about this publication:

Purpose

This publication provides information about the CT Bus clock fallback demonstration program that is included with the Intel® Dialogic® system software release.

Intended Audience

This publication is written for the following audience:

- Distributors
- System Integrators
- Toolkit Developers
- Independent Software Vendors (ISVs)
- Value Added Resellers (VARs)
- Original Equipment Manufacturers (OEMs)

How to Use This Publication

Refer to this publication after you have installed the hardware and the Intel® Dialogic® system software that includes the CT Bus clock fallback demo program.

This guide assumes that you are familiar with the Linux* operating system and the C++ programming language. You should also have a thorough understanding of CT Bus clocking concepts (primary, secondary and network reference clock masters).

Information in this guide is organized as follows:

- Chapter 1, “Demo Description” provides a brief overview of the CT Bus clock fallback demo.
- Chapter 2, “System Requirements” discusses the hardware and software required to run the demo.

- Chapter 3, "Preparing to Run the Demo" lists the procedures you must follow before running the demo.
- Chapter 4, "Running the Demo" describes procedural information for starting, using and stopping the demo.
- Chapter 5, "Demo Details" provides a table that lists the files used by the demo.

Related Information

Refer to the following documents and Web sites for more information:

- *OA&M API Library Reference*
- *OA&M API Programming Guide*
- *System Release for Linux Administration Guide*
- *Intel ® DM3 Architecture Products on Linux Configuration Guide*
- *High Availability Demo Guide*
- *Release Guide*
- *Release Update*
- *http://developer.intel.com/design/telecom/support/* (for technical support)
- *http://www.intel.com/design/network/products/telecom/index.htm* (for product information)
- *http://www.ectf.org/* (for CT Bus clocking specifications)

Demo Description 1

This chapter describes the capabilities of the CT Bus clock fallback demo program.

The CT Bus clock fallback demo program illustrates how to use the C++ classes, including the event notification framework, of the OA&M API to develop a system clocking daemon that controls clocking agents on a board and provides clock fallback when a CT Bus clocking fault or T1/E1 network alarm occurs.

The CT Bus clock fallback demo program provides the following features:

- designating a board as the CT Bus primary clock master
- designating a board as the CT Bus secondary clock master
- configuring a board to drive the clocking signal on the CT Bus network reference line
- CT Bus clock fallback to a new clock master if the primary or secondary clock masters fail
- CT Bus clock fallback to a new network reference board if the board driving clocking signal on the network reference line fails
- reception of T1/E1 network alarms

System Requirements 2

This chapter describes the requirements for running the CT Bus clock fallback demo. Topics include:

2.1 Hardware Requirements

Your system must contain a minimum of two Intel ® DM3 Architecture boards (a primary master and a secondary master) to run the CT Bus clock fallback demo. However, the system should ideally include at least three Intel® DM3 Architecture boards to see the full capabilities of the demo (primary/secondary line fallback and network reference line fallback).

2.2 Software Requirements

To run the CT Bus clock fallback demo, you need the Intel® Dialogic® system release on Linux operating systems.

Preparing to Run the Demo 3

This section includes procedural information that must be followed before running the CT Bus clock fallback demo. Topics include the following:

3.1 Editing Configuration Files

The CT Bus clock fallback demo program requires you to create a configuration file that defines the TDM bus settings for your system (bus clock rate, primary/secondary masters, network reference (Netref) master etc.). The configuration file is then passed as a command line argument when the demo is started according to the procedure in Section 4.1, "Starting the Demo", on page 15.

Notes: ***1.*** You must save the created file with a *.cfg* file extension.

2. Any board in the system that does not appear in the TDM bus configuration file is designated as a Slave.

3. If you do not create a TDM bus configuration file, the demo will use the */tmp/clapi_demo.cfg* file.

The following example file illustrates the format of the configuration file you must create:

```
# EXAMPLE CONFIGURATION FILE:
#
# (All lines beginning with # are comments)
#

# Set TDM bus type. Can be H100, H110
TDMBusType  : H110

# Set clock rates in KHz. Can be 2, 4 or 8. Is almost always set to 8
BUSCR                : 8
GROUP1CR             : 8
GROUP2CR             : 8
GROUP3CR             : 8
GROUP4CR             : 8

# Set Primary Line. Can Be A or B
PrimaryLines         : A

# Determine primary master board. Any valid integer (use pbl utility to determine which values
#apply)
PrimaryPCIBus        : 0
PrimaryPCISlot       : 14

# set primary clock rate in KHz. Usually 8 (for 8KHZ which is clockrate for both sources:
#NETREF and OSC)
PrimaryRate          : 8
```

```
# Determine source of Primary Clock. Can be NETREF or OSC
PrimarySource         : OSC

# Rate of Primary clock in KHz (must be 8)
PrimaryRate           : 8

# Determine the secondary master (use pbl utilility to determine which values apply)
SecondaryPCIBus       : 0
SecondaryPCISlot      : 15
SecondaryRate         : 8
SecondarySource       : OSC


# Set Netref board. Any valid integer (use pbl to determine which values apply)
NetrefPCIBus          : 0
NetrefPCISlot         : 16

# Set source for NetRef line. Can be 1 through number of trunks on the board
NetrefSource          : 1

# Netref clock rate in KHz. Can only be set to 8
NetrefRate            : 8

# Determine Netref fallback board
NetrefFallbackPCIBus  : 0
NetrefFallbackPCISlot : 12
NetrefFallbackSource  : OSC
NetrefFallbackRate    : 8
```

3.2 Disabling the System Software Clocking Daemon

You must disable the default clocking daemon that is included with the system software before running the CT Bus clock fallback demo. Use the procedures outlined in the *Intel® DM3 Architecture on Linux Configuration Guide* to set the **ClockDaemonMode** parameter in the *dlgsys.cfg* file to DISABLED before issuing the `dlstart` utility.

Running the Demo 4

This chapter includes information about using the CT Bus clock fallback demo. The following topics are discussed:

4.1 Starting the Demo

You must run the *cdaemon_demo* file located at */usr/dialogic/demos/ha_demos/clockapidemo* along with any command line options as specified in Section 4.2, "Demo Options", on page 15 to start the demo.

4.2 Demo Options

The command line options available when running the CT Bus clock fallback demo allow you to indicate the TDM bus configuration file the demo will read (as created according to the procedures in Section 3.1, "Editing Configuration Files", on page 13), set the debug level for log file output purposes, determine whether or not the demo runs in the foreground or as a daemon and instruct the demo to ignore duplicate clocking events.

Table 1 lists all the command line switches for the CT Bus clock fallback demo:

Table 1. Command Line Switches

Switch	Valid Values	Description
--debug <level> or -d<level>	0-5	Determines the debug level that is written to the log file. 0 is no logging output while 5 creates an entry in the log for every entry and exit from each function within the demo
--help or -h	--	Invokes the demo help screen
--file<filename> or -f<filename>	complete path to configuration file (e.g. /usr/dialogic/cfg/clapi_demo.cfg)	Indicates which TDM bus configuration file the demo should use. If a file is not supplied, the /temp/clapi_demo.cfg file is used as the default.
--foreground or -F	--	Runs the demo program in the foreground instead of in the back ground as a daemon
--timeout or -t		Sets a timeout for any duplicate clocking events

4.3 Using the Demo

This section contains information about using the demo to perform clock fallback based on user-generated clocking faults (simulated by stopping a board with the `stopbrd` utility) and T1/E1 alarms (simulated by removing a T1/E1 trunk from one of the board's network interfaces). The following topics are included:

- Primary Master Fallback
- Network Reference Master Fallback

4.3.1 Primary Master Fallback

The following procedure shows how to initiate primary clock master fallback with the CT Bus clock fallback demo. This procedure assumes you have three boards in your system that are configured via the TDM bus configuration file as follows:

Board1
: Configured as the primary master and using the internal oscillator (OSC) as its source.

Board2
: Configured as the secondary master and using the internal oscillator (OSC) as its source.

Board3
: Has no entry in the TDM bus configuration file, therefore is designated as a slave.

Use the following procedure to initiate primary clock master fallback:

1. Use the `stopbrd` utility on Board1 (the primary master). This simulates a failure on the primary clock master.

2. The demo detects that the primary master is no longer active and promotes Board2 to the primary master role and Board3 to the secondary master (assuming Board2 is capable of being a clock master).

3. Invoke the `tblist` utility located at */usr/dialogic/bin* to verify the new TDM bus configuration. The `tblist` utility outputs the current TDM bus configuration for each board in the system. The utility can be used to determine which boards are assigned primary master, secondary master and netref master roles.

   Refer to the *System Release for Linux Administration Guide* for information about the `tblist` utility.

4.3.2 Network Reference Master Fallback

The following procedure shows how to initiate network reference (Netref) line clock fallback with the CT Bus clock fallback demo. This procedure assumes you have three boards in your system that are configured via the TDM bus configuration file as follows:

Board1
: Configured as the primary master and using the netref line as its source.

Board2
: Configured as the secondary master and using the netref line as its source. This board is also configured as the netref fallback board using the T1/E1 trunk plugged into its second network interface as the netref fallback source.

Board3
: Configured as the netref master and using the T1/E1 trunk plugged into its first network interface as the netref source.

Use the following procedure to initiate netref clock master fallback:

1. Remove the T1/E1 trunk from the first network interface of Board3 (i.e the netref source). This simulates a network alarm on the T1/E1 line.

2. The demo detects that a signal is no longer present on the netref source and promotes the netref fallback board (Board2) to the netref master. The netref line source is now the T1/E1 trunk that is plugged into the second network interface of Board2.

3. Invoke the `tblist` utility located at *usr/dialogic/bin* to verify the new TDM bus configuration. The `tblist` utility outputs the current TDM bus configuration for each board in the system. It can be used to determine which boards are assigned primary master, secondary master and netref master roles.

4.4 Stopping the Demo

Use the Linux `kill` command to stop the CT Bus clock fallback demo.

Demo Details 5

This chapter provides more details about the CT Bus clock fallback demo program.

5.1 Files Used by the Demo

Table 2 lists the files used by the CT Bus clock fallback demo. All files are located in the */usr/dialogic/demos/ha_demos/clockapidemo* directory.

Table 2. Files Used by the CT Bus Clock Fallback Demo

File Name	Purpose
ClkDmonMain.cpp	Primary demo program. Defines how the demo starts
CAgent.cpp	Code that manages the individual clocking agents on a board
CBoard.cpp	Code that manages a board
CBus.cpp	Code that manages all clocking agents on the TDM bus
CConfigFile.cpp	Code that allows the demo to read the user-defined TDM bus configuration file
CEventConsumer.cpp	Network alarm and clocking fault event handler
CLog.cpp	Defines a log interface. Logs output to the Linux syslog or optionally to a user-designated file
ClkDmonErrorCodes.h	Header file with #defines for error codes
CFallbackHandler.cpp	Fallback logic for network alarms
cdaemon_demo	CT bus clock fallback demo executable
Makefile.demo	Makefile for compiling the demo
tdmbus.cfg	Example configuration file

Index