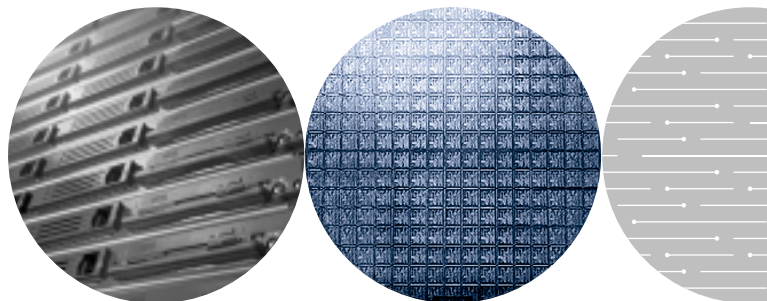




Using Amtelco Multi-Chassis Interconnect Boards with Intel® Telecom Boards

Intel in
Communications



Contents

Introduction	1
The Issue	1
System Configuration	2
Installing the Boards	2
Configuring the CT Bus Using DCM	3
Configure the Intel® Board(s)	3
Configure the Amtelco Board	3
Configuring the Clocking Properties of the CT Bus	6
CT Bus Clock Fall-Back	6
Starting the System	6
Starting Order	6
Configure the Amtelco Boards	7
Sharing CT Bus Timeslots Across Servers	9
Starting the System	9
Receiving Messages from the Amtelco Board	11

Figures

Figure 1: Simple System Diagram	2
Figure 2: Fiber Cable Connection for a System with Two Amtelco Boards	3
Figure 3: Add Device Dialog	3
Figure 4: Name Third-Party Board Dialog	4
Figure 5: Select Third-Party Device Capabilities	4
Figure 6: Third-Party Device Time slot Allocation Dialog	5
Figure 7: DCM Showing Added Third-Party Device	5
Figure 8: TDM Bus Properties Dialog	6
Figure 9: TDM Bus Clock Settings	6
Figure 10: Code for Configuring the Amtelco Board	7
Figure 11: Amtelco – CT Bus Fiber Ring Connection Diagram	9
Figure 12: Code to Make Amtelco and Intel CT Bus-to-Fiber Ring Connection on Server	10
Figure 13: Code for Message Retrieval from the Amtelco Board	12

Introduction

This document describes the basic setup and use of Amtelco Multi-Chassis Interconnect Boards with Intel® Dialogic® or Intel® NetStructure™ boards on the Windows* operating system using System Release 5.1.1 Feature Pack 1.

It provides basic instructions for using Amtelco and Intel boards together. The developer will need to expand on these concepts to meet the specific requirements of a particular solution. Also, this document does not discuss any interchassis communication a specific solution may require.

Please note that for the purpose of this discussion, a server is defined as a single chassis with at least one Amtelco and one Intel board. A system is a group of servers connected using an Amtelco fiber ring.

The Issue

Intel® System Release 5.1.1 Feature Pack 1 software for Windows allows multiple boards to share resources across the CT Bus. However, Intel offers no mechanism for sharing the CT Bus across servers. This functionality requires a device that can connect the CT Bus across servers. One possible solution is to use the Amtelco XDS* Series Multi-Chassis Interconnect Boards. Other options may be possible, but there are few products available today that can provide this capability without significant development work. The Amtelco product is an inexpensive and field-proven product available today.

The Multi-Chassis Interconnect Boards enable sharing of the CT Bus using a fiber ring. This permits multiple chassis to function as one, resulting in increased density and allowing the sharing of resources.

For installation instructions for the Intel and Amtelco boards, refer to the installation guides provided with your product(s).

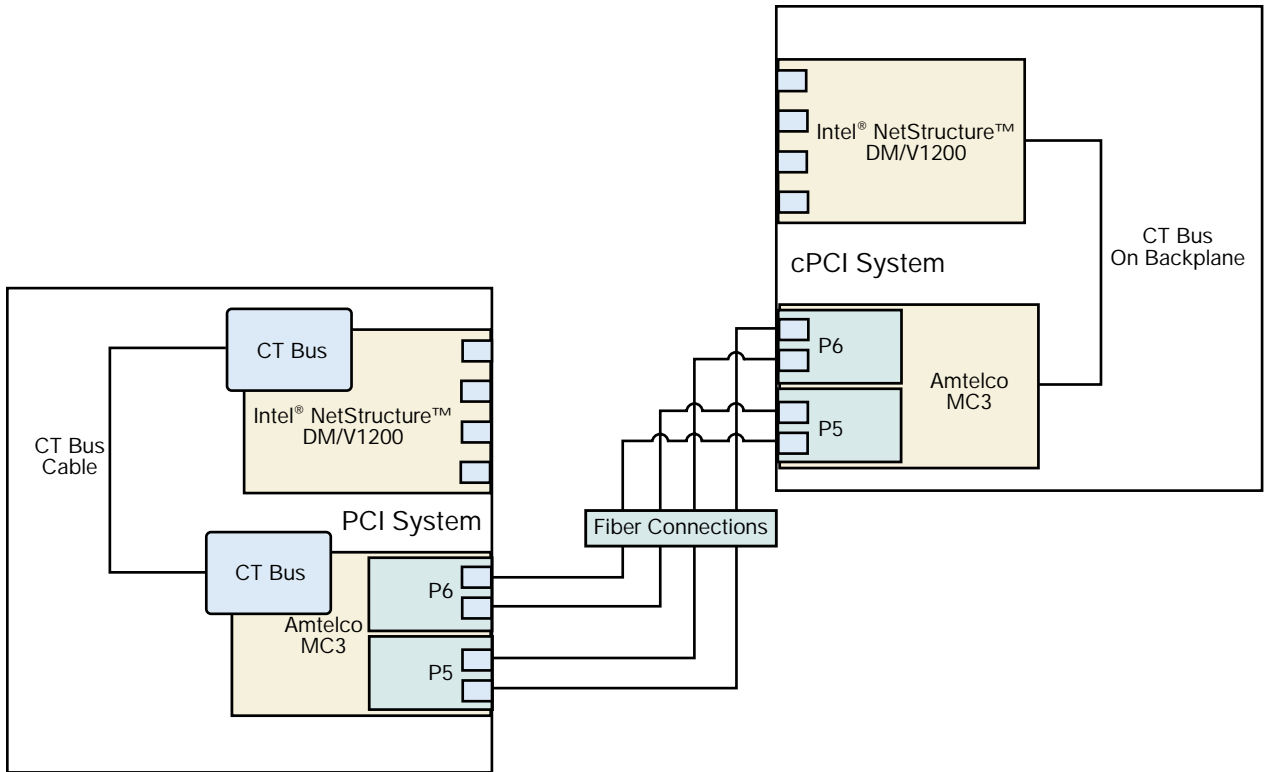


Figure 1: Simple System Diagram

System Configuration

The configuration consists of at least two servers. One server is configured as the master and one or more are configured as slaves. The master server provides the source of the clock for the entire system via the fiber ring. The slaves acquire their clock from the master through the Amtelco fiber ring. For the master server, the Amtelco board needs to be configured as a CT Bus clock slave in the Intel® Dialogic® Configuration Manager (DCM). In the slave system(s), the Amtelco board is configured as the CT Bus clock master in DCM.

Installing the Boards

Begin by installing the boards according to the installation guide provided with each board. Note that on a PCI system, all Intel and Amtelco boards installed in the system need to be interconnected using a CT Bus cable. Failure to interconnect the boards will result in no communication between the Intel and Amtelco boards. This issue does not arise on the CompactPCI* platform because the CT Bus is part of the backplane and requires no additional cables.

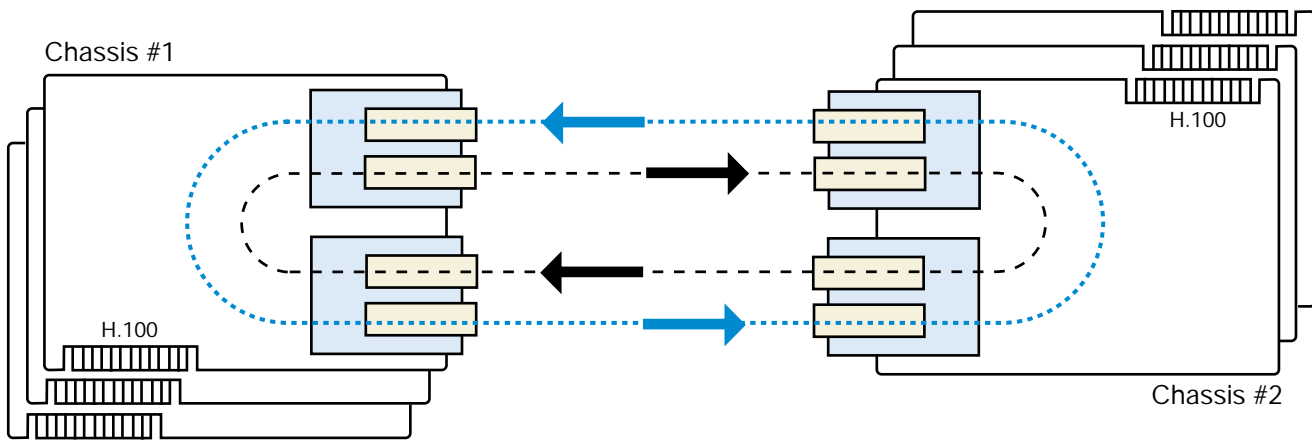


Figure 2: Fiber Cable Connection for a System with Two Amtelco Boards

After all the servers are configured in the system, interconnect the Amtelco boards using standard fiber cables with SC connectors. In a system with two servers, connect the cables as shown in Figure 2. To increase the number of servers, just break one of the P5-P6 cable pairs and insert another machine linking the P5 from chassis 1 to P6 of the new chassis, and from P5 of the new chassis to P6 of chassis 2. Systems can be located up to 2,000 meters apart.

Configuring the CT Bus Using DCM

Configure the Intel® Board(s)

Start the Intel Dialogic Configuration Manager (DCM) and configure the Intel boards according to the documentation provided with each board.

Configure the Amtelco Board within DCM

In DCM, select Add Device from the Action Menu. The following dialog appears:

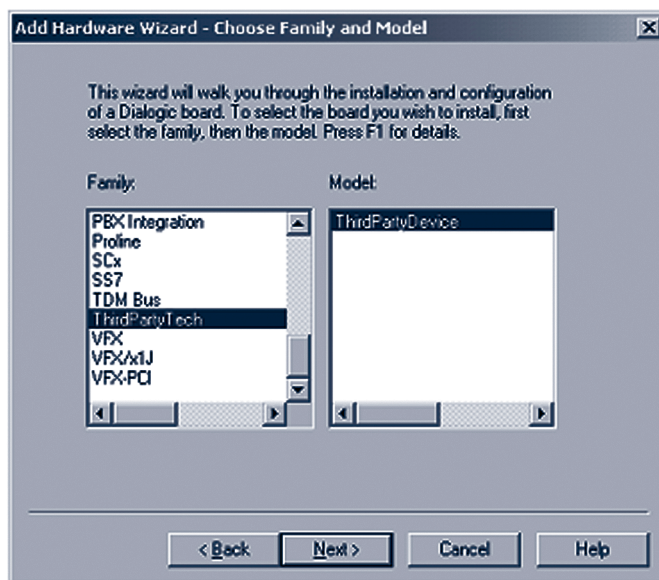


Figure 3: Add Device Dialog

Select ThirdParty from the Family List and ThirdPartyDevice from the Model List, then click Next.

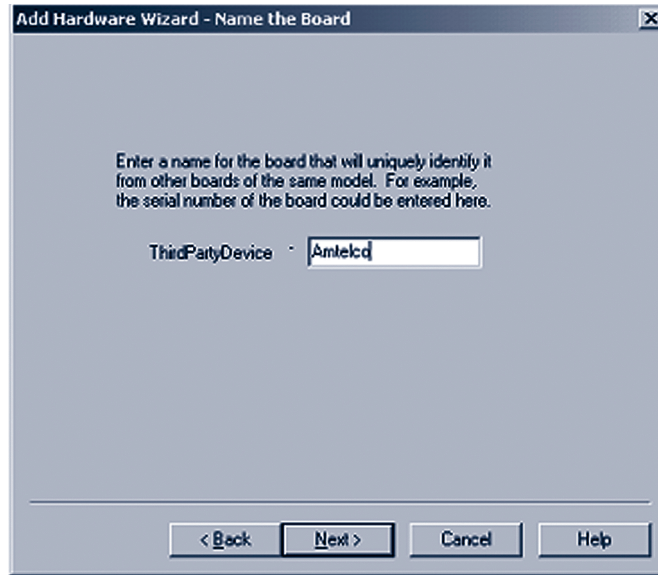


Figure 4: Name Third-Party Board Dialog

Next, enter a name for the device you are configuring (e.g., Amtelco) and click Next.

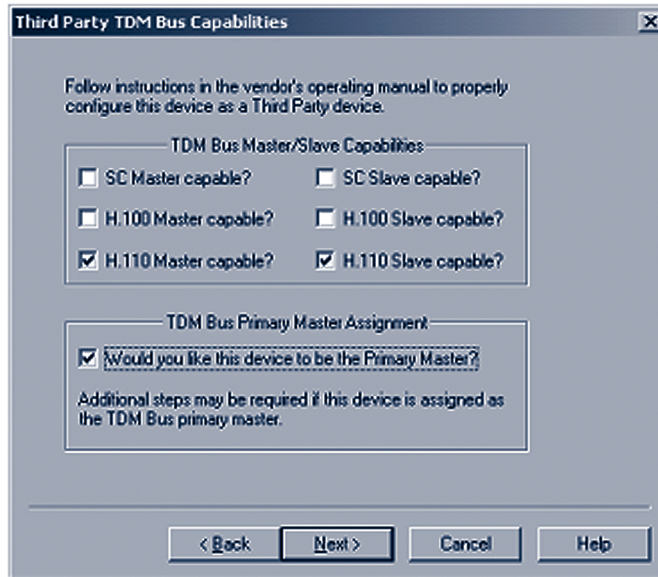


Figure 5: Select Third-Party Device Capabilities

Now select the appropriate TDM Bus capabilities. You need to select both primary and slave entries for the appropriate type (for PCI, select H.100; for CompactPCI, select H.110). If this device will provide the primary clock for the CT Bus (slave server), check the TDM Bus Primary Master Assignment, then click Next. Figure 5 shows the selections for a CompactPCI server that is a slave in the system.

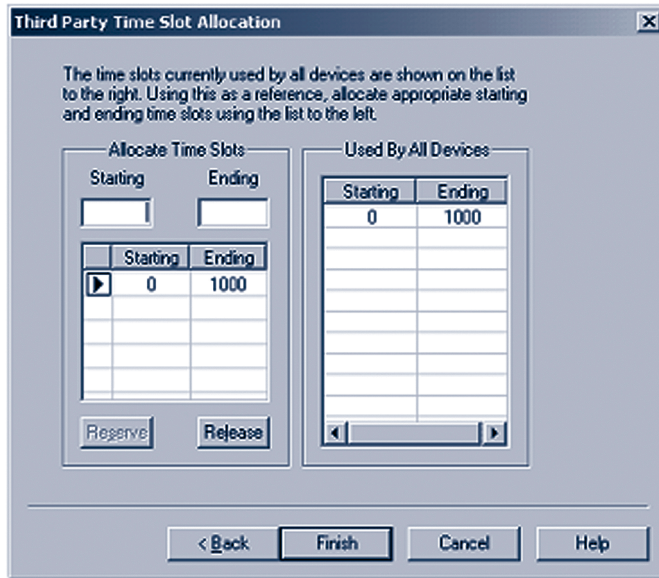


Figure 6: Third-Party Device Time Slot Allocation Dialog

The dialog in figure 6 is used to reserve time slots for use by the Amtelco board. You can reserve any range between 0 and 4095 for use by the Amtelco board. DCM adjusts the Intel board timeslots around the ranges provided. If too many timeslots are reserved for use by the Amtelco board, leaving too few for the installed Intel board(s), the Intel software will be unable to complete startup.

It is also possible to dynamically allocate time slots using the NCM API from an application. For purposes of this document, enter 0 for the minimum time slot and 1000 for the maximum time slot, then click Reserve. The range should now appear in the list.

Note that you need to allow for the timeslot requirements of the Amtelco board. For example, if five DMN160TEC boards are installed in a chassis running a full density of 480 channels per board, 2400 timeslots will be required for just the Intel boards. This leaves only 1696 timeslots for use by the Amtelco board, limiting the number of full duplex connections to 1696. So, therefore, if you need full duplex connections for all available channels, the limit would be four DMN160TEC boards, or 1920 channels.

To complete the configuration, click Finish. DCM now displays the third-party device as shown in figure 7.

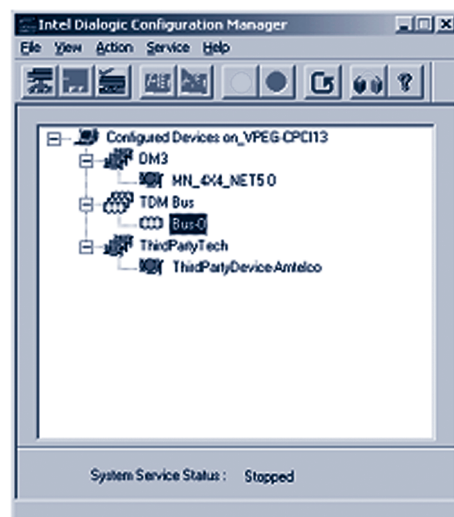


Figure 7: DCM Showing Added Third-Party Device

Configuring the Clocking Properties of the CT Bus

After the Amtelco board is configured, the clocking information for the server is correctly configured. The following information is provided for reference and verification of the settings. To verify or change the clocking information, right-click on the TDM Bus device in DCM and select Properties.

Figure 8 shows a server configured as a slave of the system. The Amtelco board is configured as the primary master, so the clock is taken from the fiber ring and placed onto the CT Bus for use by the other boards in the server

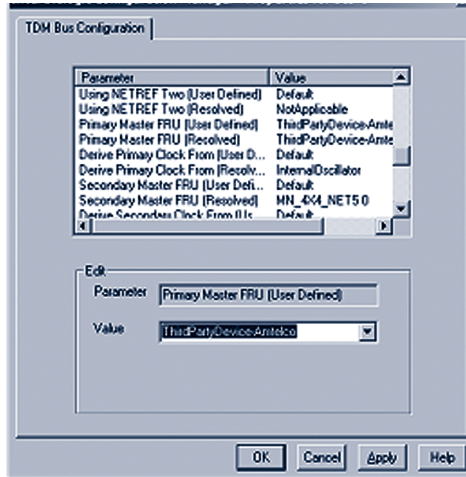


Figure 8: TDM Bus Properties Dialog

Figure 9 lists the clock settings for the TDM Bus for both the master and slave servers of the system.

Server Designation	Amtelco CT Bus Setting	Primary Master FRU (User-Defined)	Secondary Master FRU (User-Defined)
Slave	Master	Amtelco device	Intel device
Master	Slave	Intel device	Intel device

Figure 9: TDM Bus Clock Settings

CT Bus Clock Fallback

Clock fallback is the mechanism used to restore the CT Bus clock source in the event of a clocking fault or a T-1 or E-1 network alarm.

In the configuration described in this document, if the master server fails, the CT Bus clock provided by the Amtelco board on the slave system provides an internally oscillated clock to the CT Bus. As a result, the clock on the CT Bus becomes out of sync with the public network, rendering the fiber ring unavailable. To restore the ring, restart the master server. If this is not possible, one stopped one of the slaves and reconfigured it as the master, then restart. Currently, real-time clock fallback configuration capabilities are only available on Intel, Dialogic, System Release 6.0 for Linux*, which offers an operations, administration, and management (OA&M) API that allows dynamic changing of the CT Bus clock configuration.

Starting the System

Starting Order

The start order of the servers in the system is crucial. The master server of the system must be started first since it provides the clocking source to the MC3 fiber ring. If the master server is not started before the other systems in the ring, there is no clock source and the other systems immediately fall back to the secondary clock master. Once the system has fallen back to the secondary clock master, the fiber ring, it is not available for use by the application because the MC3 clock is not in synch with the CT Bus clock.

You can start systems in any order; however, the fiber ring of the Amtelco boards is not available until the master server is up and running.

Configure the Amtelco boards

Before starting DCM, you must execute an application to configure the Amtelco card using the Amtelco API to provide and obtain the clocking information for the CT Bus and fiber ring and to configure the mode in which the Amtelco fiber ring will operate. The fiber ring can operate in either extended mode (4846 timeslots) or redundant mode (2423 timeslots).

Figure 10 shows an example of code that configures an Amtelco board. This application would need to be executed on each server in the system.

```
int main(int argc, char* argv[])
{
    if (argc < 3)
    {
        printf("Usage AmtelcoConfig board#, Master-Slave\r\n");
        printf("\tboard# - The Amtelco board #\r\n");
        printf("\tMaster-Slave 1 for Master - 0 for Slave\r\n");
        return 0;
    }

    unsigned char boardNumber = (unsigned char) atoi(argv[1]);
    bool bMaster = (argv[2][0] == '1') ? true : false;

    if (xds_board_reset(boardNumber) != 0)
    {
        printf("Error Resetting board #s\r\n", argv[1]);
        return 0;
    }

    if (xds_mc3_ring_mode(boardNumber, 0) != 0)
    {
        printf("Error Setting ring mode for board #s\r\n", argv[1]);
        return 0;
    }

    //the following is for PCI only if using CPCI comment out
    if (g_bServerIsPci)
    {
        if (xds_set_ct_rate(boardNumber, 2, 2, 2, 2) != 0)
        {
            printf("Error Setting CT stream rates for board #s\r\n", argv[1]);
            return 0;
        }
    }
    if (xds_set_encoding(boardNumber, 'A') != 0)
    {
        printf("Error Setting encoding to Alaw for board #s\r\n", argv[1]);
        return 0;
    }

    printf("Setting board #s to %s\n", argv[1],
        (const char*)(bMaster ? "Master" : "Slave"));
    if (bMaster)
        nRet = xds_set_ct_clock_mode(boardNumber, 2, 3, 0, 0, 0);
    else
        nRet = xds_set_ct_clock_mode(boardNumber, 1, 0, 0, 0, 0);
    printf("%s\n", (const char*)(nRet == 0) ? "Succeded" : "Failed");

    return 0;
}
```

Figure 10: Code for Configuring the Amtelco Board

Following are brief descriptions of the APIs used in the configuration application. The complete API description is in the Amtelco programming guide.

xds_set_encoding

Sets encoding for the Amtelco board.

```
xds_set_encoding(BoardNumber, enc);
```

BoardNumber – Amtelco board number

Enc – Encoding 'A' for A-Law 'M' for Mu-Law

xds_set_ct_clock_mode

Set the clock information for the Amtelco board.

```
xds_set_ct_clock_mode(BoardNumber,
ClockMode, SubMode,
Arg1, Arg2, Arg3);
```

BoardNumber – Amtelco board number

Clock Mode	Sub Mode	Arg 1	Arg 2-Arg 3
1 - clocks is obtained from CT bus (slave)	0 – Obtain Clock from CT-Bus clock A	0 – No CT_NETREF output	0 - unused
2- the board is clock master CT bus clock A	3 – Obtain Clock from Ring 0 or Ring 1	0 or 1 – Obtain clock from Ring n	0 - unused

Note: Not all possible values have been documented here.

xds_mc3_ring_mode

Set the operating mode for the Amtelco boards Fiber Ring.

```
xds_mc3_ring_mode (BoardNumber, mode);
```

BoardNumber – Amtelco board number

mode = ring mode

0 - extended mode, 4846 timeslots

1 - redundant mode, primary ring is 0, 2423 timeslots

2 - redundant mode, primary ring is 1, 2423 timeslots

Starting the System

Start DCM on the Intel server that was chosen to be the master of the system. Once the master server is running, all slaves can be simultaneously started. Now click the green start button (●) on the toolbar of DCM. The server attempts to start the services for DCM. When the service starts, the board indicators next to each item in DCM show green.

Sharing CT Bus Timeslots Across Servers

For time slot sharing, link the local CT Bus on one server to the Amtelco fiber ring, then connect from the fiber ring to the local CT Bus on another server. Figure 11 is a simple diagram of this process, showing a link from a media device (dxxxBxCx) on server 1 to another media device (dxxxByCy) on server 2.

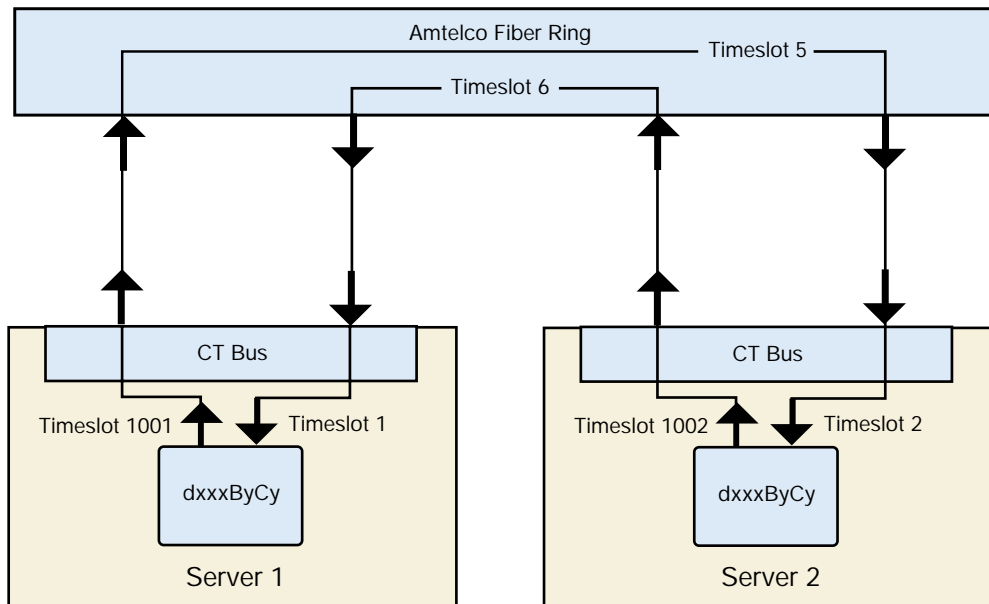


Figure 11: Amtelco/Intel CT Bus-to-Fiber-Ring Connection Diagram

The use of the time slots by the Amtelco board is assigned using the `xds_mc3_connect`, `xds_mc3_listen`, and `xds_h100_listen` functions. The `xds_mc3_connect` function creates a full duplex connection between the fiber ring and the CT Bus. All parameters for this function are viewed from the Amtelco board. The CT Bus Transmit parameter is where the Amtelco board will transmit data onto the CT Bus.

Figure 12 is an example of the code needed to make the connection in the diagram on server 1. Note that `xds_mc3_listen` and `xds_h100_listen` functions do not view the connections in the same way. Be sure to check the API manual before using these functions.

```
//hVox is the handle of a previous dx_open for "dxxxxBxCx"

int mc3_Xmit, mc3_Rcv;
if (!getFiberRingTimeSlotsForDevice(hVox, &mc3_Xmit, &mc3_Rcv))
{
    // unable to allocate TimeSlots on Fiber Ring
    return;
}
long ctbus_Rcv;
if (!getAvailableCTBusTimeSlotForAmtelco(&ctBus_Rcv))
{
    // unable to get available CTBus Slot
    // release Fiber Ring Timeslots for Fiber Ring back to manager
    return;
}
SC_TSINFO tsi;
long ctbus_Xmit;
tsi.sc_numts = 1;
tsi.sc_tsarrayp = &ctbus_Xmit;
if (dx_getxmitslot(hVox, &tsi) == -1)
{
    // Error Retrieving Xmit Slot
    // Release Fiber Ring Timeslots for Fiber Ring back to manager
    return;
}
// To align with Figure 11 the following values would have been
returned
// mc3_Xmit = 5 and mc3_Rcv = 6
// ctbus_Xmit = 1002 ctbus_Rcv = 2

// Make the CT-Bus connection
SC_TSINFO tdiListen;
long ltsiListen = (long) mc3_Xmit;
tsiListen.sc_numts = 1;
tsiListen.sc_tsarrayp = &ltsiListen;
if (dx_listen(hVox, &ltsiListen) == -1)
{
    // Listen Failed
    // Release Fiber Ring Timeslots for Fiber Ring back to manager
    return;
}
// Make the Fiber Ring connection
xds_mc3_connect(BoardNumber, ctbus_Rcv, ctbus_Xmi
```

Figure 12: Code to Make Amtelco and Intel CT Bus – Fiber Ring Connection on Server 1

Receiving Messages from the Amtelco Board

The Amtelco boards use messages to communicate with the application. This is primarily for reporting clocking and ring errors to the application.

The Amtelco board does not send messages to the application by default, so the application needs to call the `xds_msg_on` function to request that messages be sent. When the application is shutting down, it needs to call `xds_msg_off` to stop the board from sending messages. (Note: If more than one running application monitors the messages from the board, be sure that only the last running application calls `xds_msg_off`.)

The application retrieves messages from the board by calling `xds_msg_receive`. Waiting for messages should be done in a separate thread of your application so that response to errors can be as quick as possible. The `xds_msg_receive` function is non-blocking, so Amtelco has an event object that can be used to block the thread until a message is received. This prevents the thread from using 100 percent of the CPU.

To obtain the handle to the event object, the application would call the Amtelco API `OpenEventHandle`. The implementation of this function, installed by Amtelco as source code, needs to be included in the build of the application.

Figure 13 shows the basic method for retrieving messages from a single Amtelco board. This function is called from a thread created by the application, terminating only when the `TerminateThread` function is called.

```

bool bEndMessageLoop = false;
HEVENT hEvent = NULL;
void MsgThreadProc(unsigned char BoardNumber)
{
    if (OpenEventHandle(&hEvent) == 0)
        return; // Event Handle Available not available
    XDS_RCVMSG xrReceivedData;
    memset(&xrReceivedData, 0, sizeof(XDS_RCVMSG));
    xds_msg_on(BoardNumber);
    while (!bEndMessageLoop)
    {
        while (xds_msg_receive(&xrReceivedData) == SUCCESS)
        {
            // Identify and Process Message Here
        }
        WaitForSingleObject(hEvent, INFINITE);
    }
    xds_msg_off(BoardNumber);
    hEvent = NULL;
}

void TerminateThread()
{
    if (hEvent)
    {
        bEndMessageLoop = TRUE;
        SetEvent(hEvent);
    }
}

xds_msg_receive
Turn off messaging from the Amtelco board
xds_msg_off(XDS_RCVMSG*);
    XDS_RCVMSG* - Pointer to allocated structure

typedef struct rcvmsg
{
    unsigned char board_number; /* board number/SW1 setting */
    char msg_type; /* message type */
    char msg_sub; /* message subtype */
    int port_number; /* port number of message */
    int timeslot; /* timeslot of message */
    unsigned char argument; /* argument of message */
    char msg_str[32]; /* String Argument of Message */
    unsigned short augRxLen; /* Receive Length */
    unsigned char augRxMsg[260]; /* Receive Buffer */
}
XDS_RCVMSG;



---


xds_msg_on
Turn on messaging from the Amtelco board
xds_msg_on(BoardNumber);
    BoardNumber - Amtelco board number



---


xds_msg_off
Turn off messaging from the Amtelco board
xds_msg_off(BoardNumber);
    BoardNumber - Amtelco board number

```

Figure 13: Code for Message Retrieval from the Amtelco Board

To learn more, visit our site on the World Wide Web at <http://www.intel.com>

1515 Route Ten
Parsippany, NJ 07054
Phone: 1-973-993-3000
Fax: 1-973-993-3093

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, or life sustaining applications. Intel may make changes to specifications and product descriptions at any time, without notice.

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

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, reference <http://www.intel.com/procs/perf/limits.htm> or call (U.S.) 1-800-628-8686 or 1-916-356-3104.

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

