PCI ITU/CEPT E1 Demultiplexer

Model INT-9951

15 March 2001

Intronics, Incorporated
2020 Lafayette Boulevard
Fredericksburg, Va 22401

(540) 898-4518 (Voice)
(540) 898-3603 (Fax)
http://www.intrnx.com

INTRONICS, INC.

MODEL INT-9951

PCI ITU / CEPT E1 Demultiplexer

15 March 2001

Photographs on this Page.

Table of Contents

1.0 Overview

The INT-9951 is a demultiplexer for ITU/CEPT telecommunications standard E1 rate (2.048 Mbps) PCM signals. It is a single standard length PCI Bus printed circuit board that is compatible with many host computers and software operating systems. Software drivers are available for Windows NT/2000, Solaris (SPARC), Solaris 8 (Intel), Linux and Irix. Multiple INT-9951 demultiplexers may be installed in a single host computer, limited only by the number of free PCI slots available.

The E1 demultiplexer includes a 31 x 31 non-blocking crosspoint switch. This feature allows the user to assign any E1 PCM input timeslot (1 – 31) to any one or more of the 31 audio outputs. It permits applications such as activity scanning, output routing based on signal type recognition, and many others.

2.0 Functional Description

To facilitate integration with a wide variety of E1 sources the INT-9951 accepts all of the following input signal types:

1. AMI/HDB3 conforming to ITU/CCITT standard G.703.
2. ECL NRZ data and clock conforming to MECL 10K or 100K (single ended or differential pair).
3. LVDS NRZ data and clock conforming to IEEE 1394.
4. EIA RS-422 NRZ data and clock.
5. PECL NRZ data and clock.
6. TTL NRZ data and clock.

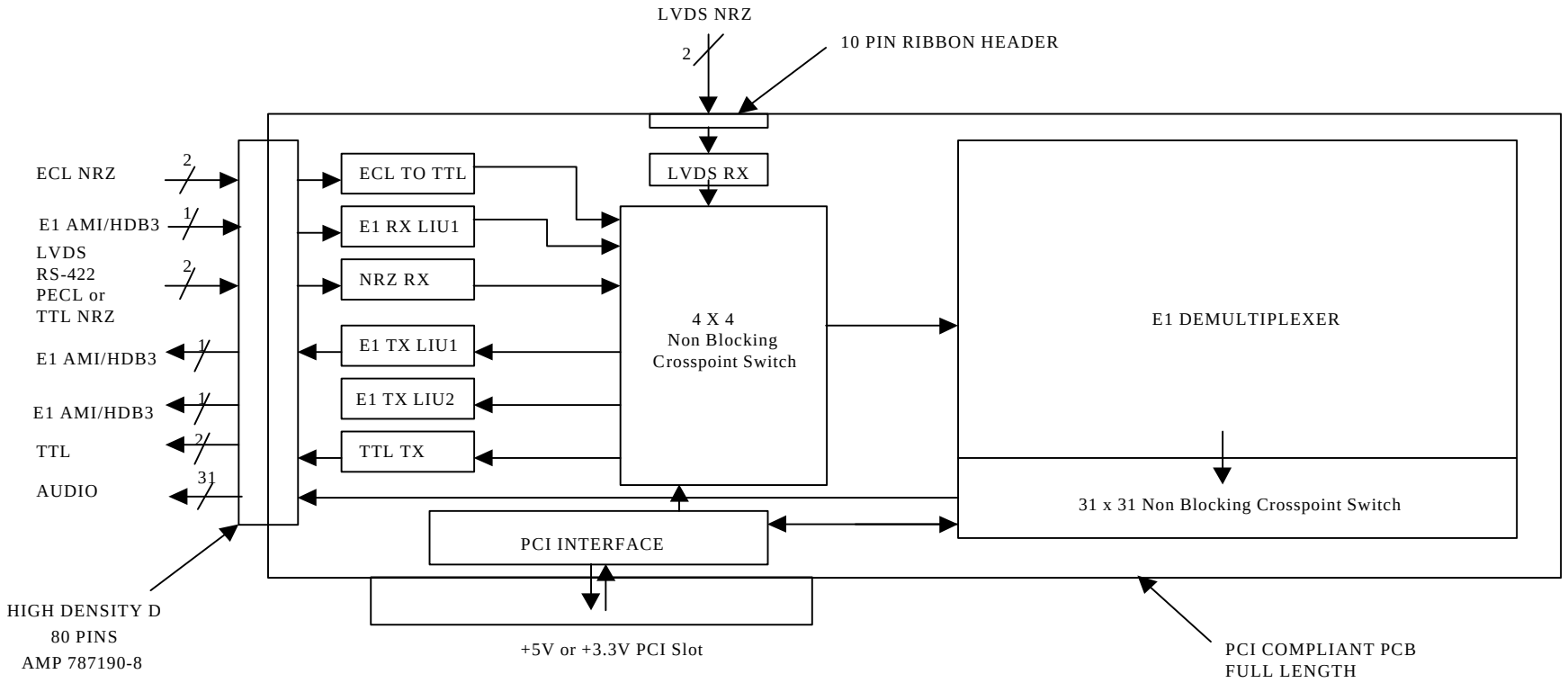
The LVDS, EIA RS-422, PECL and TTL inputs are optional and only one may be selected per PCB.

In addition to serving as the source for the E1 demultiplexer, any of the inputs listed above may be selected as the source for the following outputs:

1. Two channels of AMI/HDB3 conforming to ITU/CCITT standard G.703.
2. TTL NRZ data and clock.

These features allow the unit to serve both as an E1 demultiplexer and as a signal format converter. For example the LVDS input may be routed to both the E1 demultiplexer and the two AMI/HDB3 outputs, while simultaneously routing the ECL input to the TTL output.

A functional block diagram of the INT-9951 is shown on the following page:



INT-9951 PCI ITU/CEPT E1 DEMULTIPLEXER

FUNCTIONAL BLOCK DIAGRAM

15 March 2001

3.0 Hardware Elements

The INT-9951 is comprised of a single PCI standard length multi-layer printed circuit card. It requires only +5V power from the PCI bus and is compatible with both +5V and +3.3V PCI bus host interface slots.

3.1 External Signal Interface

All user signal interfaces to the INT- 9951 are accommodated via a single 80-pin rear edge panel connector. See Appendix A for a list of pin assignments for this connector.

3.2 Internal Configuration Options

The INT-9951 provides no user selectable internal on card configuration options. LED D1 is located at the top left edge of the card. It is illuminated at power on while the Xilinx FPGA is initializing. This process takes about one second, after which the LED is turned off. If the LED remains on this indicates a hardware failure as the Xilinx FPGA has failed to load. The PCI interface will not function and the board should be returned for repair.

3.3 Preventive Maintenance

There are no routine maintenance procedures. No periodic alignments or adjustments are required.

4.0 Software

There are three layers of software provided with the INT-9951. These are the PCI interface control functions, the application program interface (API), and the host control application program (HCP).

4.1 PCI Interface Control Functions

The INT-9951 PCI interface is a licensed design, provided by The Engineering Design Team, Inc. (EDT) of Beaverton Oregon, and represented in their PCI-16D product. A copy of the software for the PCI-16D is provided with the INT-9951 and must be installed by the user to provide the PCI interface driver functions. The API provided with the INT-9951 uses two of these functions to communicate with the card via the PCI bus. Programmers wishing to develop programs for the INT-9951 need not directly use any of the PCI interface control functions, but should instead use the API provided. The PCI interface driver functions used are edt_open(), and edt_ioctl(). Refer to the EDT User's Guide for the PCI 16D if additional details of these functions are required.

4.2 Application Program Interface (API)

The host control application program provided with the INT-9951 makes use of the API and provides a comprehensive menu driven user interface. This will be adequate for many users and no additional software will be required. Often however the user will want to integrate the control of the INT-9951 into his own software environment. Development of customized applications by the user for the INT-9951 is made possible through the use of the API. The API is written in the 'C' programming language and source code is provided. The API is found within the source file named "CMD_9951.C". The API calls functions found in the source file named "L9951.C" which perform low-level operations using the PCI interface drivers provided by EDT. Developers of customized applications need only use the API function described below, and not the low-level functions found in "L9951.C". Source code of the host control application program is also provided and should serve as a convenient example of the use of the API.

The API is a single 'C' language function named cmd_9951(). It accepts one argument that is a pointer to an ASCII command string, and returns a pointer to an ASCII results string. The paragraphs below describe each command string and all results strings associated with that command. An unrecognized command string is discarded by the API, and a pointer to the results string "ERR! INVALID COMMAND" is returned.

4.2.1 PCI

Function:	Open a PCI interface channel to an INT-9951. This must precede all other commands.
Command string:	"PCI x" where x is the PCI unit number 0 – 9 parameter
Response strings:	"ERR! Invalid PCI unit number parameter" "ERR! Unable to open INT-9951 PCI Interface" "" null string, command was successful
Example usage:	cmd_response = cmd_9951("PCI 0");

4.2.2 GET INIT_FLAG

Function:	Determine if INT-9951 has previously been initialized for operation after initial power on.
Command string:	"GET INIT_FLAG"
Response strings:	"INITIALIZED" "NOT INITIALIZED"
Example usage:	cmd_response = cmd_9951("GET INIT_FLAG");

4.2.3 INIT

Function: Initialize for operation after power on. Must be performed after PCI command and before all other commands except GET INIT_FLAG when power is first applied to the unit.

Command string: "INIT"

Response strings: "" null string, command was successful

Example usage: cmd_9951("INIT");

Default Settings: The following default settings are commanded:

DEMUX_IN	LIU1_RX
TTL_OUT	ECL_NRZ_IN
LIU1_OUT	LIU1_RX
LIU2_OUT	LIU1_RX

Audio Outputs 1–31 connected to Demux Time slots 1-31

4.2.4 VER

Function: Get manufacturer and software version.

Command string: "VER"

Response string: "Intronics, Inc. Model INT-9951 E1 Demux 22 Feb 01 Ver 1.02"

Example usage: cmd_response = cmd_9951("VER");

4.2.5 CPYRT

Function: Get copyright notice.

Command string: "CPYRT"

Response string: "Copyright © 2001, Intronics, Inc. ALL RIGHTS RESERVED"

Example usage: cmd_response = cmd_9951("CPYRT");

4.2.6 SET DEMUX_IN

Function: Select the input source for the E1 demultiplexer.

Command string: "SET DEMUX_IN xxxxxxx", where xxxxxxx may be: LIU1_RX, REAR_NRZ, TOP_NRZ or ECL_NRZ

Response strings: "ERR! Invalid signal source"
"" null string, command was successful

Example usage: cmd_response = cmd_9951("SET DEMUX_IN LIU1_RX");

4.2.7 GET DEMUX_IN

Function: Get the currently selected input source for the E1 demultiplexer.

Command string: "GET DEMUX_IN"

Response strings: "LIU1_RX"
"REAR_NRZ"
"TOP_NRZ"
"ECL_NRZ"

Example usage: cmd_response = cmd_9951("GET DEMUX_IN");

4.2.8 SET TTL_OUT

Function: Select the input source for the TTL output.

Command string: "SET TTL_OUT xxxxxxx", where xxxxxxx may be: LIU1_RX, REAR_NRZ, TOP_NRZ or ECL_NRZ

Response strings: "ERR! Invalid signal source"
"" null string, command was successful

Example usage: cmd_response = cmd_9951("SET TTL_OUT ECL_NRZ");

4.2.9 GET TTL_OUT

Function: Get the currently selected input source for the TTL output.

Command string: “GET TTL_OUT”

Response strings: “LIU1_RX”
“REAR_NRZ”
“TOP_NRZ”
“ECL_NRZ”

Example usage: cmd_response = cmd_9951(“GET TTL_OUT”);

4.2.10 SET LIU1_OUT

Function: Select the input source for the E1 LIU1 output.

Command string: “SET LIU1_OUT xxxxxxx”, where xxxxxxx may be:
LIU1_RX, REAR_NRZ, TOP_NRZ or ECL_NRZ

Response strings: “ERR! Invalid signal source”
“” null string, command was successful

Example usage: cmd_response = cmd_9951(“SET LIU1_OUT LIU1_RX”);

4.2.11 GET LIU1_OUT

Function: Get the currently selected input source for the E1 LIU1 output.

Command string: “GET LIU1_OUT”

Response strings: “LIU1_RX”
“REAR_NRZ”
“TOP_NRZ”
“ECL_NRZ”

Example usage: cmd_response = cmd_9951(“GET LIU1_OUT”);

4.2.12 SET LIU2_OUT

Function: Select the input source for the E1 LIU2 output.

Command string: "SET LIU2_OUT xxxxxxx", where xxxxxxx may be: LIU1_RX, REAR_NRZ, TOP_NRZ or ECL_NRZ

Response strings: "ERR! Invalid signal source"
"" null string, command was successful

Example usage: cmd_response = cmd_9951("SET LIU2_OUT LIU1_RX");

4.2.13 GET LIU2_OUT

Function: Get the currently selected input source for the E1 LIU2 output.

Command string: "GET LIU2_OUT"

Response strings: "LIU1_RX"
"REAR_NRZ"
"TOP_NRZ"
"ECL_NRZ"

Example usage: cmd_response = cmd_9951("GET LIU2_OUT");

4.2.14 TFSYN

Function: Get the primary E1 framing synchronization status from the demultiplexer.

Command string: "TFSYN"

Response strings: "LOCKED"
"UNLOCKED"

Example usage: cmd_response = cmd_9951("TFSYN");

4.2.15 MFSYN

Function: Get the E1 multi-frame synchronization status from the demultiplexer.

Command string: "MFSYN"

Response strings: "LOCKED"
"UNLOCKED"

Example usage:
```
cmd_response = cmd_9951( "MFSYN" );
```

4.2.16 AIS

Function: Get the all ones alarm status from the demultiplexer.

Command string: "AIS"

Response strings: "ON"
"OFF"

Example usage:
```
cmd_response = cmd_9951( "AIS" );
```

4.2.17 SET CONN

Function: Connect a demultiplexer audio output to an E1 input PCM timeslot.

Command string: "SET CONN xxx yyy"
xxx is the audio output number 1 – 31, or ALL
yyy is the E1 input timeslot number 1 – 31 or DEFAULT

Response strings: "ERR! Audio output parameter not in range (1 – 31)"
"ERR! Audio output parameter not 1 – 31 or ALL"
"ERR! E1 input timeslot parameter not in range (1 – 31)"
"ERR! E1 input timeslot parameter not 1 – 31 or DEFAULT"
"" null string, command was successful

Example usage:
```
cmd_response = cmd_9951( "SET CONN  8 18" );
cmd_response = cmd_9951( "SET CONN ALL 14" );
cmd_response = cmd_9951( "SET CONN 17 DEFAULT" );
cmd_response = cmd_9951( "SET CONN ALL DEFAULT" );
```

4.2.18 GET CONN

Function:	Get the E1 input PCM timeslot currently connected to a demultiplexer audio output.
Command string:	"GET CONN xxx" xxx is the audio output number 1 – 31
Response strings:	"ERR! Audio output parameter not a number" "ERR! Audio output parameter not in range (1 – 31)" "TS-xx" where xx is timeslot 01 – 31"
Example usage:	cmd_response = cmd_9951("GET CONN 8");

4.3 HOST CONTROL APPLICATION PROGRAM (HCP)

The host control application program (HCP) provided with the INT-9951 makes use of the API and provides a comprehensive menu driven user interface. It provides all the features normally required to operate the INT-9951. The HCP is written in the 'C' programming language and source code is provided. The HCP is found within the source file named "M9951.C". The following paragraphs describe the features of the HCP and its operation.

4.3.1 STARTUP

The HCP is a main applications program executable under a wide variety of operating systems on any hardware platform supporting the PCI bus. When the program is started it executes the following steps:

1) Displays the manufacturer, address, copyright notice and software version.
2) Opens the PCI interface for communications with the INT-9951.
3) Tests the INT-9951 to see if it has already been initialized. If it has then the HCP does **not** repeat the initialization (resetting the INT-9951 to it's default operating state). This allows the user to terminate the HCP and later restart it without affecting the operational configuration of the INT-9951. If the INT-9951 has not already been initialized then it is set to its default operating state and an initialization message is displayed. Refer to the INIT command in the API section above for these defaults.
4) Displays a HELP menu of all available commands.
5) Displays a command prompt and awaits a user command input.

4.3.2 HELP MENU COMMAND

The HCP HELP menu is displayed upon initial startup and may be displayed by the user by entering the command “help” at the command prompt.

The command HELP results in the following:

> help

```
INIT              - Initialize
INFO              - Display Manufacturer and Software Version
SET DEMUX_IN      - Select E1 Demux Input Source
SET TTL_OUT       - Select TTL Output Source
SET LIU1_OUT      - Select E1 Line Interface Unit 1 Output Source
SET LIU2_OUT      - Select E1 Line Interface Unit 2 Output Source
SET CONN          - Map Audio Output to E1 Input Timeslot
STATUS            - Display System Status
SAVE              - Save current unit setup in a file
RESTORE           - Restore unit setup from a file
EXIT              - Terminate Program
HELP              - Help

      Enter HELP .. for HELP about a specific command
      Where  ..  represents  the  command  characters
      Example: HELP STATUS is help for STATUS display
```

4.3.3 INIT COMMAND

```
INIT - Initialize the INT-9951 E1 Demultiplexer. Performs a
       hardware reset and restores all settings to their
       factory defaults. These settings are as follows:

       SET DEMUX_IN LIU1_RX
       SET TTL_OUT ECL_NRZ
       SET LIU1_OUT LIU1_RX
       SET LIU2_OUT LIU1_RX
       SET CONN ALL DEFAULT (Audio Out = E1 Timeslot In, 1-1, 2-2, etc.)
```

4.3.4 INFO COMMAND

```
INFO - Display manufacturer name, address, web site and the
       date and version of this software
```

4.3.5 SET DEMUX_IN COMMAND

```
SET DEMUX_IN - Select E1 demultiplexer input source

       Format: SET DEMUX_IN xxx where xxx is the source

       xxx - LIU1_RX  - E1 AMI/HDB3 coded rear panel connector input
             REAR_NRZ - NRZ data and clock rear panel connector input
             TOP_NRZ  - LVDS NRZ data and clock top edge connector input
             ECL_NRZ  - ECL NRZ data and clock rear panel connector in

Note: REAR_NRZ is configured at the factory for TTL, RS-422, PECL, or
LVDS

       Example command: SET DEMUX_IN REAR_NRZ
```

4.3.6 SET TTL_OUT COMMAND

```
SET TTL_OUT - Select TTL Output source

       Format: SET TTL_OUT xxx where xxx is the source

       xxx - LIU1_RX  - E1 AMI/HDB3 coded rear panel connector input
             REAR_NRZ - NRZ data and clock rear panel connector input
             TOP_NRZ  - LVDS NRZ data and clock top edge connector input
             ECL_NRZ  - ECL NRZ data and clock rear panel connector in

Note: REAR_NRZ is configured at the factory for TTL, RS-422, PECL, or
LVDS

       Example command: SET TTL_OUT ECL_NRZ
```

4.3.7 SET LIU1_OUT COMMAND

```
SET LIU1_OUT - Select E1 Line Interface Unit 1 Output source

       Format: SET LIU1_OUT xxx where xxx is the source

       xxx - LIU1_RX  - E1 AMI/HDB3 coded rear panel connector input
             REAR_NRZ - NRZ data and clock rear panel connector input
             TOP_NRZ  - LVDS NRZ data and clock top edge connector input
             ECL_NRZ  - ECL NRZ data and clock rear panel connector in

Note: REAR_NRZ is configured at the factory for TTL, RS-422, PECL, or
LVDS

       Example command: SET LIU1_OUT LIU1_RX
```

4.3.8 SET LIU2_OUT COMMAND

```
SET LIU2_OUT - Select E1 Line Interface Unit 2 Output source

       Format: SET LIU2_OUT xxx where xxx is the source

       xxx - LIU1_RX  - E1 AMI/HDB3 coded rear panel connector input
             REAR_NRZ - NRZ data and clock rear panel connector input
             TOP_NRZ  - LVDS NRZ data and clock top edge connector input
             ECL_NRZ  - ECL NRZ data and clock rear panel connector in

Note: REAR_NRZ is configured at the factory for TTL, RS-422, PECL, or
LVDS

       Example command: SET LIU2_OUT LIU1_RX
```

4.3.9 SET CONN COMMAND

```
SET CONN - Assign E1 Input Stream Timeslot to an Audio Output

       Format: SET CONN xxx yyy

       xxx - Audio output (1 - 31) or ALL
       yyy - E1 input stream timeslot (1 - 31) or DEFAULT
             DEFAULT sets input timeslot to output stream

       Example commands: SET CONN 4 11
                         SET CONN ALL 17
                         SET CONN 17 DEFAULT
                         SET CONN ALL DEFAULT
```

4.3.10 STAT COMMAND

```
STATUS - Display System Status

       Format: STATUS

       Displays the following information:

           Demux Input Source    (selected by SET DEMUX_IN)
           TTL Output Source     (selected by SET TTL_OUT)
           E1 LIU1 Output Source (selected by SET LIU1_OUT)
           E1 LIU2 Output Source (selected by SET LIU2_OUT)

           E1 Frame Synchronization Status (LOCKED or UNLOCKED)
           E1 All Ones Alarm Status        (OFF or ON)

           Table of all Audio Outputs and the E1 Input Timeslots
           assigned to them (selected by SET CONN)
```

4.3.11 SAVE COMMAND

SAVE - Saves current unit settings to a file

Format: SAVE xxx where xxx is the file pathname

An ASCII text file is created containing commands identical to the keyboard commands required to setup the unit in its current state. The file extension .TXT is added to the end of the pathname (xxx) entered. The file created may be edited by any text editor if desired.

4.3.12 RESTORE COMMAND

RESTORE - Restores unit settings from a file

Format: RESTORE xxx where xxx is the file pathname

An ASCII text file is read containing commands that must be identical to the keyboard commands required to setup the unit to the desired state. The file extension .TXT is added to the end of the pathname (xxx) entered. The file may have been created using the SAVE command, or by any text editor.

4.3.13 EXIT COMMAND

EXIT - Terminates this program

Format: EXIT

Note: This program may be terminated and restarted without affecting the current setup and operation of the unit. The only time the program initializes the unit is the first time it is run after power has been turned on. After that the program may be terminated and restarted and will not reinitialize the unit.

5.0 Specifications

User Signal Interface Connector

Number of Pins	80
PCB Connector	AMP P/N 787190-8
PCB Mating Connector	AMP P/N 749111-7
Straight Shielded Backshell	AMP P/N 749196-1
Right Angle Backshell	AMP P/N 749205-1
Pin Assignments	See Appendix A

User Input Signal Definitions

Signal Name	E1_IN_TIP, E1_IN_RING
Rate	2048 Kbps
Format	ITU G.703, G.704
Coding	AMI/HDB3
Impedance	120 Ohms Balanced Differential

Signal Name	ECL_IN_DATA, ECL_IN_DATA/ ECL_IN_CLOCK, ECL_IN_CLOCK/
Format	MECL 10KH Logic Levels
Coding	NRZ
Impedance	100 Ohms Balanced Differential Each Signal
Clock	Rising Edge Centered on Data Bit

Signal Name	LVDS_IN_DATA, LVDS_IN_DATA/ LVDS_IN_CLOCK, LVDS_IN_CLOCK/
Format	IEEE 1394
Coding	NRZ
Impedance	100 Ohms Balanced Differential Each Signal
Clock	Rising Edge Centered on Data Bit

User Output Signal Definitions

Signal Name	E1_OUT1_TIP, E1_OUT1_RING
Rate	2048 Kbps
Format	ITU G.703, G.704
Coding	AMI/HDB3
Impedance	120 Ohms Balanced Differential

Signal Name	E1_OUT2_TIP, E1_OUT2_RING
Rate	2048 Kbps
Format	ITU G.703, G.704
Coding	AMI/HDB3
Impedance	120 Ohms Balanced Differential

Signal Name	TTL_OUT_DATA, TTL_OUT_DATA/ TTL_OUT_CLOCK, TTL_OUT_CLOCK/
Format	TTL Logic Levels
Coding	NRZ
Impedance	Un-terminated, MC74HCT241 TTL bus driver
Clock	Rising Edge Centered on Data Bit

Signal Name	AUDIO_OUT_1 through AUDIO_OUT_31
Rate	300 Hz – 3400 Hz
Format	Analog, +/- 2.5V maximum
Impedance	10K Ohms Un-Balanced Single Ended
Source	Selectable from any demultiplexed E1 timeslot

PCI Interface

Specifications	PCI Local Bus Specification, Revision 2.2
Size	Full length, single slot
Slot Type	+5V or +3.3V
Bus Clock	33 MHz maximum
Software Supported	WinNT, Win/2000, Solaris (SPARC), Solaris 8 (Intel), Linux, Irix

Physical

Size	Full length PCI card, (12.25” x 4.20” x 0.60”)
PCI Slots Required	One
Weight	12 Oz
Power	7 Watts (nominal) (1.5A @ +5VDC)
Operating Temp	0 – 50 degrees C (Non-Condensing)
Storage Temp	-20 – 60 degrees C (Non-Condensing)

Appendix A – Edge Connector Pin Assignments

AMP 749111-7	STANDARD RIBBON P1	SIGNAL		AMP 749111-7	STANDARD RIBBON P2	SIGNAL
1	1	AUDIO_OUT_1		41	1	AUDIO_OUT_17
2	2	GND		42	2	GND
3	3	AUDIO_OUT_2		43	3	AUDIO_OUT_18
4	4	GND		44	4	GND
5	5	AUDIO_OUT_3		45	5	AUDIO_OUT_19
6	6	GND		46	6	GND
7	7	AUDIO_OUT_4		47	7	AUDIO_OUT_20
8	8	GND		48	8	GND
9	9	AUDIO_OUT_5		49	9	AUDIO_OUT_21
10	10	GND		50	10	GND
11	11	AUDIO_OUT_6		51	11	AUDIO_OUT_22
12	12	GND		52	12	GND
13	13	AUDIO_OUT_7		53	13	AUDIO_OUT_23
14	14	GND		54	14	GND
15	15	AUDIO_OUT_8		55	15	AUDIO_OUT_24
16	16	GND		56	16	GND
17	17	AUDIO_OUT_9		57	17	AUDIO_OUT_25
18	18	GND		58	18	GND
19	19	AUDIO_OUT_10		59	19	AUDIO_OUT_26
20	20	GND		60	20	GND
21	21	AUDIO_OUT_11		61	21	AUDIO_OUT_27
22	22	GND		62	22	GND
23	23	AUDIO_OUT_12		63	23	AUDIO_OUT_28
24	24	GND		64	24	GND
25	25	AUDIO_OUT_13		65	25	AUDIO_OUT_29
26	26	GND		66	26	GND
27	27	AUDIO_OUT_14		67	27	AUDIO_OUT_30
28	28	GND		68	28	GND
29	29	AUDIO_OUT_15		69	29	AUDIO_OUT_31
30	30	GND		70	30	GND
31	31	AUDIO_OUT_16		71	31	GND
32	32	GND		72	32	ECL_IN_DATA/
33	33	LVDS_IN_DATA/		73	33	ECL_IN_DATA
34	34	LVDS_IN_DATA		74	34	ECL_IN_CLOCK/
35	35	LVDS_IN_CLOCK/		75	35	ECL_IN_CLOCK
36	36	LVDS_IN_CLOCK		76	36	E1_OUT2_RING
37	37	TTL_OUT_CLOCK		77	37	E1_OUT2_TIP
38	38	TTL_OUT_DATA		78	38	GND
39	39	E1_OUT1_RING		79	39	E1_IN_RING
40	40	E1_OUT1_TIP		80	40	E1_IN_TIP