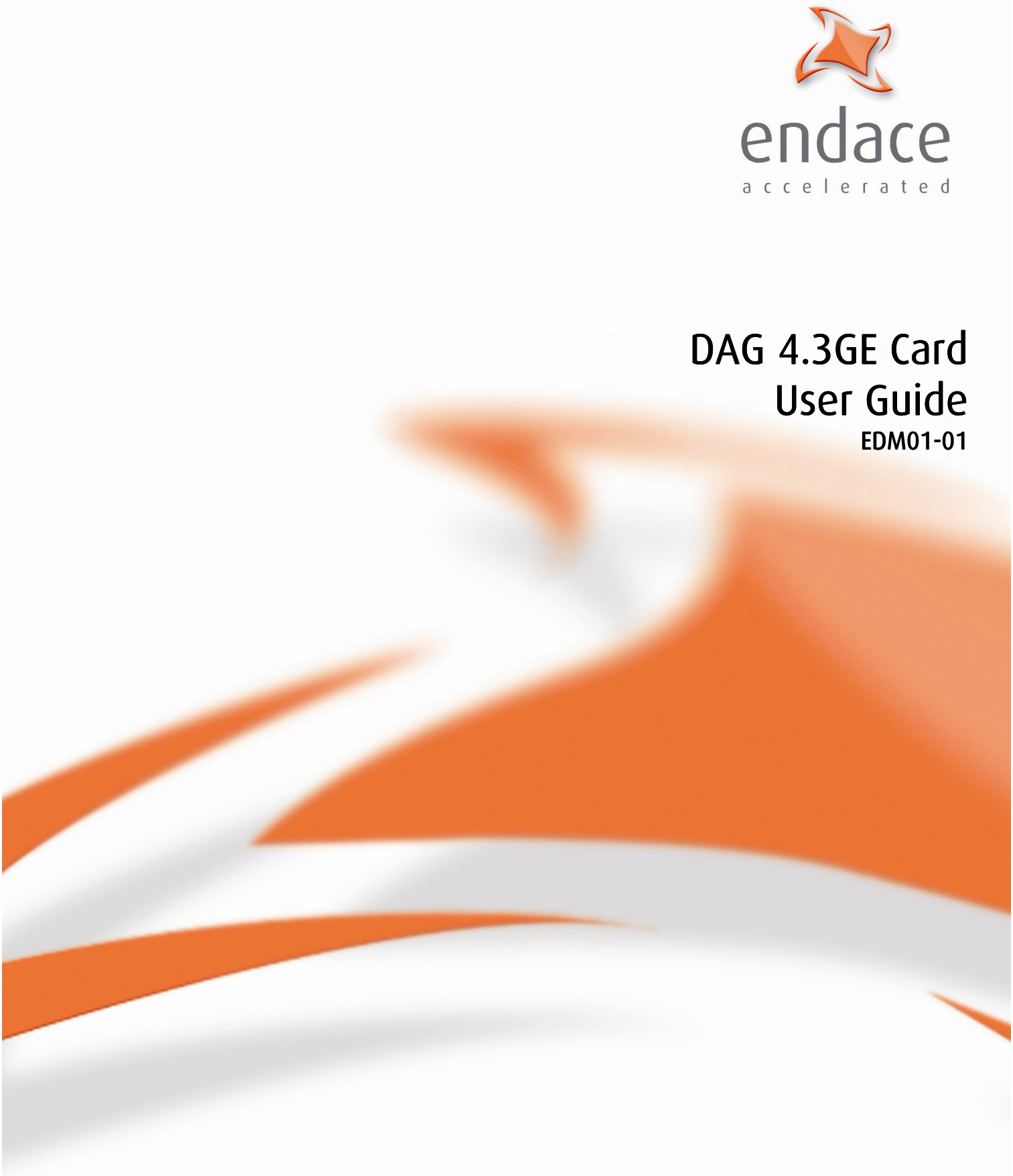

DAG 4.3GE Card User Guide

EDM01-01

www.endace.com

Published by:

Endace Measurement Systems® Ltd

Building 7
17 Lambie Drive

PO Box 76802
Manukau City 1702
New Zealand

Phone: +64 9 262 7260

Fax: +64 9 262 7261

support@endace.com

www.endace.com

International Locations

New Zealand
Endace Technology® Ltd

Level 9
85 Alexandra Street
PO Box 19246
Hamilton 2001
New Zealand

Phone: +64 7 839 0540
Fax: +64 7 839 0543

Americas
Endace USA® Ltd

Suite 220
11495 Sunset Hill Road
Reston
Virginia 20190
United States of America

Phone: ++1 703 382 0155
Fax: ++1 703 382 0155

Europe, Middle East & Africa
Endace Europe® Ltd

Sheraton House
Castle Park
Cambridge CB3 0AX
United Kingdom

Phone: ++44 1223 370 176
Fax: ++44 1223 370 040

Published by:

Endace Measurement Systems® Ltd

Building 7
17 Lambie Drive

PO Box 76802
Manukau City 1702
New Zealand

Phone: +64 9 262 7260

Fax: +64 9 262 7261

support@endace.com

www.endace.com

International Locations

New Zealand
Endace Technology® Ltd

Level 9
85 Alexandra Street
PO Box 19246
Hamilton 2001
New Zealand

Phone: +64 7 839 0540
Fax: +64 7 839 0543

Americas
Endace USA® Ltd

Suite 220
11495 Sunset Hill Road
Reston
Virginia 20190
United States of America

Phone: ++1 703 382 0155
Fax: ++1 703 382 0155

Europe, Middle East & Africa
Endace Europe® Ltd

Sheraton House
Castle Park
Cambridge CB3 0AX
United Kingdom

Phone: ++44 1223 370 176
Fax: ++44 1223 370 040

Protection Against Harmful Interference

When present on equipment this manual pertains to, the statement "This device complies with part 15 of the FCC rules" specifies the equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the Federal Communications Commission [FCC] Rules.
These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment.
This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications.
Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

Extra Components and Materials

The product that this manual pertains to may include extra components and materials that are not essential to its basic operation, but are necessary to ensure compliance to the product standards required by the United States Federal Communications Commission, and the European EMC Directive. Modification or removal of these components and/or materials, is liable to cause non compliance to these standards, and in doing so invalidate the user's right to operate this equipment in a Class A industrial environment.

Disclaimer

Whilst every effort has been made to ensure accuracy, neither Endace Measurement Systems Limited nor any employee of the company, shall be liable on any ground whatsoever to any party in respect of decisions or actions they may make as a result of using this information.

Endace Measurement Systems Limited has taken great effort to verify the accuracy of this manual, but assumes no responsibility for any technical inaccuracies or typographical errors.

In accordance with the Endace Measurement Systems policy of continuing development, design and specifications are subject to change without notice.

Table of Contents

Chapter 1 Introduction

Introduction The installation of the Endace DAG 4.3GE card on a PC begins with installing the operating system and the Endace software. This is followed by fitting the card and connecting the ports.

Viewing this document This document, DAG 4.3GE Card User Manual is available on the installation CD.

In this chapter This chapter covers the following sections of information.

- User Manual Purpose
- DAG 4.3GE Card Product Description
- DAG 4.3GE Card Architecture
- DAG 4.3GE Card Extended Functions
- DAG 4.3GE Card System Requirements

1.1 User Manual Purpose

Description The purpose of this DAG 4.3GE Card User Manual is to describe:

- Installing DAG 4.3GE Card
- Setting Optical Power
- Confidence Testing
- Running Data Capture Software
- Synchronizing Clock Time
- Data Formats Overview

Pre-requisite This document presumes the DAG card is being installed in a PC already configured with an operating system.

A copy of the Debian Linux 3.1 (Sarge) is available as a bootable ISO image on one of the CD's shipped with the DAG card.

To install on the Linux/FreeBSD operating system, follow the instructions in the document EDM04.05-01r1 Linux FreeBSD Installation Manual, packaged in the CD shipped with the DAG card.

To install on a Windows operating system, follow the instructions in the document EDM04.05-02r1 Windows Installation Manual, packaged in the CD shipped with the DAG card

1.2 DAG 4.3GE Card Product Description

Description The DAG 4.3GE card is a PCI-X bus card designed for cell and packet capture and generation on Ethernet networks.

Figure 1-1. DAG 4.3GE PCI-X Card.

The DAG 4.3GE card collects packet header and payload from Ethernet networks and is protocol independent. Full packet capture at line rate allows recording of all header information and/or payload with a high precision timestamp.

The DAG 4.3GE is capable of transmitting packets at 100% line rate on both ports while simultaneously receiving packets at 100% line rate on both ports.

1.3 DAG 4.3GE Card Architecture

Description Serial Ethernet optical network data received by two 1000baseSX optical interfaces flow into an Ethernet Framer ASIC then immediately into the Field-Programmable Gate Array [FPGA].

The FPGA contains an Endace DAG Universal Clock Kit [DUCK] timestamp engine, packet record processor, and PCI-X interface logic.

Because of component close association, packets or cells are time-stamped accurately. Time stamped packet records are stored in an external FIFO memory before transmission to the host.

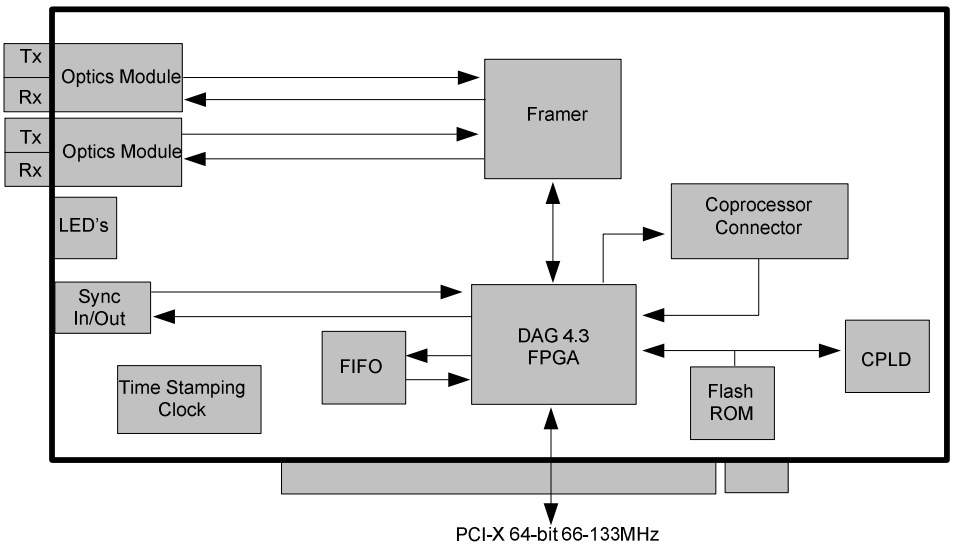

Figure 1-2. DAG 4.3GE Card Major Components and Data Flow.

1.4 DAG 4.3GE Card Extended Functions

Description

The DAG 4.3GE card is equipped with a coprocessor connector which can be used with the optional Endace DAG Coprocessor as a data processing tool.

The IP packet classification specifications are:

- Packets are classified by TCP/IP header fields and/or payload content.
- Up to 16,384 TCP/IP header classification rules. Up to ~2,000 arbitrary payload search strings.
- Classification rules are assigned a user-defined 16-bit identifier
- Packets matching classification rules are assigned the matching rule's identifier.
- Programmable actions may be associated with each rule identifier. For example, The packet should either be dropped, retransmitted to the network, or presented to the host.
- Packets presented to the host include the rule-match identifier in the record header.

Contact the Endace customer support team at support@endace.com to enable effective use of extended functions.

1.5 DAG 4.3GE Card System Requirements

Description

The DAG 4.3GE card and associated data capture system minimum operating requirements are:

- PC, at least Intel Xeon 1.8GHz or faster
- Intel E7500, ServerWorks Grand Champion LE/HE, or newer chip set
- 256 MB RAM
- At least one free PCI-X 1.0 slot supporting 66-133MHz operation
- Software distribution free space of 30MB

Operating system

For convenience, the Debian 3.1 [Sarge] Linux system is included on the Endace Software Install CD. Endace currently supports Windows XP, Windows Server 2000, Windows Server 2003, FreeBSD, RHEL 3.0, and Debian Linux operating systems.

Different system

For advice on using a system substantially different from that specified above, contact Endace support at support@endace.com

Chapter 2: Installation

Introduction

The DAG 4.3GE card can be installed in any free PCI-X 1.0 slot. It will operate at 66, 100, or 133MHz PCI-X mode, however it will not operate correctly in 32 or 64-bit PCI slots.

Higher speed slots are recommended for best performance.

The DAG 4.3GE should be the only device on the PCI-X bus if possible as the cards make very heavy use of PCI-X bus data transfer resources.

Although the driver supports up to four DAG cards by default in one system, due to bandwidth limitations there should not be more than one card on a single PCI-X bus.

In this section

This section covers the following topics of information.

- Installation of Operating System and Endace Software
- Insert DAG 4.3GE Card into PC
- DAG 4.3GE Card Port Connectors
- Pluggable Optical Transceivers

2.1 Installation of Operating System and Endace Software

Description

If the DAG device driver is not installed, before proceeding with the next chapter, install the software by following the instructions in EDM04-01 Linux/FreeBSD Installation Guide.

To install the software on a Windows operating system, follow the instructions in EDM04-02 Windows Installation Guide.

Go to the next chapter of information when the DAG device driver is installed.

2.2 Insert DAG 4.3GE Card into PC

Description

Inserting the DAG 4.3GE card into a PC involves accessing the PCI-X bus slot, fitting the card, and secure the bus slot screw.

Procedure

Follow these steps to insert the DAG 4.3GE card.

Step 1. Access bus Slot

Power computer down.

Remove PCI-X bus slot cover.

Procedure, continued

Step 2. Fit Card

Insert DAG 4.3GE card into PCI-X bus slot.

Step 3. Replace bus Slot Screw

Secure card with screw.

Step 4. Power Up Computer

2.3 DAG 4.3GE Card Port Connectors

Description

There are two duplex LC-type optical port connectors. Each port consists of an optical fibre transmitter and receiver.

The upper connection of each pair is for transmitting signals. They are connected only if loop-back facility is used in the DAG to daisy-chain the systems. They are also connected if a data generation program is being used.

The bottom connectors for each pair are used for the received signal.

An 8-pin RJ45 socket is used for time synchronization. This socket should never be connected to an Ethernet network or telephone line.

2.4 Pluggable Optical Transceivers

Description

Some newer versions of the DAG 4.3GE cards are available with pluggable optics. To provide compatibility with the broadest possible range of optical parameters, Endace offers the industry standard Small Form-factor Pluggable [SFP] optical transceiver on the DAG 4.3GE card.

The SFP transceiver consists of two parts:

- Mechanical chassis attached to the circuit board
- Transceiver unit which may be inserted into the chassis

The correct transceiver is chosen to suit the optical parameters of the target network installed in the chassis.

The transceiver may then be connected to the network via LC-style optical connectors.

Further information about the Pluggable Optical Transceiver is available at the Endace http://www.endace.com/dagPluggable.htm web page.

2.4 Pluggable Optical Transceivers, continued

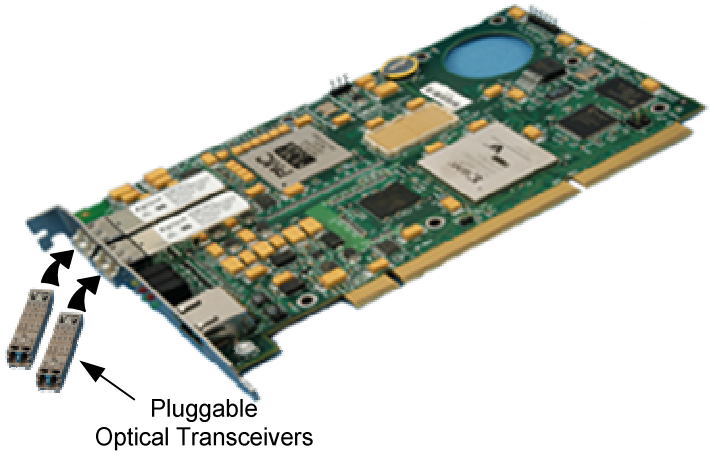

Figure 2-1. Pluggable Optical Transceivers.

Chapter 3: Setting Optical Power

Description

The optical power range depends on the particular device fitted on the DAG 4.3GE card.

The DAG 4.3GE card is shipped fitted with two 1000baseSX FTRJ 8519F 850nm multi-mode short range optics modules by default.

Optical power measure

Optical power is measured in dBm – decibels relative to 1 mW where 10 dB is equivalent to a factor of 10 in power.

The numbers are all negative, showing powers below 1 mW. The most sensitive devices can work down to about –30 dBm, or 1 uW.

Configuration

Table 3-1 shows the DAG 4.3GE card optics power module configuration.

Part #	Fibre	Data Rate	Max Power [dBm]	Min Power [dBm]	Nominal Pwr [dBm]
FTR8519F	MMF	1000	0	-22	-14

In this chapter

This chapter covers the following sections of information.

- Optical Power Input
- Splitter Losses

3.1 Optical Power Input

Description

The optical power input to DAG must be within the receiver's dynamic range of 0 to -22dBm.

When optical power is slightly out of range an increased bit error rate is experienced. If power is well out of range the system cannot lock onto the Ethernet signal. In extreme cases of being out range excess power will damage a receiver.

When power is above the upper limit the optical receiver saturates and fails to function. When power is below the lower limit the bit error rate increases until the device is unable to obtain lock and fails.

Input power

When the DAG card is set up, measure the optical power at the receiver and ensure that it is well within the specified power range.

Input power is adjusted by:

- Changing splitter ratio if power is too high or too low, or
- Inserting an optical attenuator if power is too high.

3.2 Splitter Losses

Description

Splitters have the insertion losses marked on packaging or in accompanying documentation.

- A 50:50 splitter will have an insertion loss of between 3 dB and 4 dB on each output
- 90:10 splitter will have losses of about 10 dB in the high loss output, and <2 dB in the low loss output

The 1000baseSX transceiver uses 850nm optics. Splitters used must be designed for 850nm as the insertion loss will vary for different wavelengths.

Single mode fibre loss

A single mode fibre connected to a multi-mode input has minimal extra loss.

Multi-mode fibre loss

A multi-mode fibre connected to a single mode input creates large and unpredictable loss.

Chapter 4: Confidence Testing

Introduction

The confidence testing is a process to determine the DAG 4.3GE card is functioning correctly.

The process also involves a card capture session, and demonstrates configuration in the style of 'What You See You Can Change', WYSYCC.

Interface statistics are also inspected during this process.

In this chapter

This chapter covers the following sections of information.

- Interpreting DAG 4.3GE Card LED Status
- DAG 4.3GE Card LED Display Functions
- Configuration in WYSYCC Style
- DAG 4.3GE Card Capture Session
- Inspect Interface Statistics
- Reporting Problems

4.1 Interpreting DAG 4.3GE Card LED Status

Description

The DAG 4.3GE has 8 status LEDs, one coloured blue, three green, two orange, and two red.

When a DAG 4.3GE series card is powered up the LED 1 should always come on.

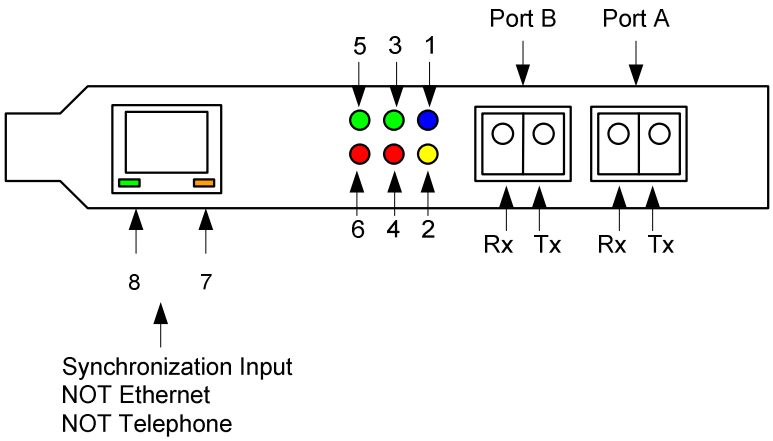

Figure 4-1. Typical DAG 4.3GE Card Status LED's.

LED definitions The following table describes the LED display definitions:

LED	Description
LED 1	FPGA successfully programmed.
LED 2	Data capture in progress.
LED 3	Port A Signal Detect – valid optical signal seen by optical receiver.
LED 4	Port A Link Error.
LED 5	Port B Signal Detect – valid optical signal seen by optical receiver.
LED 6	Port B Link Error.
LED 7	PPS Out: Pulse Per Second Out – indicates card is sending a clock synchronization signal.
LED 8	PPS In: Pulse Per Second In – indicates card is receiving an external clock synchronization signal.

4.2 DAG 4.3GE Card LED Display Functions

Description The function of the DAG 4.3GE card LED displays include indication of optical power status, packet capture activity, links on ports A and B, and PPS signals.

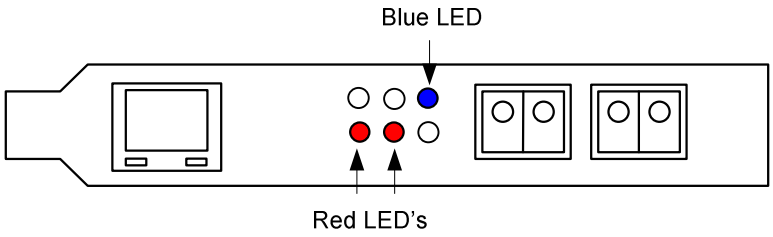

Figure 4-2. LED State for DAG 4.3GE Card Without Optical Input.

4.3 Configuration in WYSYCC Style

Description

Configuration in WYSYCC is the 'What You See You Can Change' style.

Running the command '`dagfour`' alone shows the current configuration. Each of the items displayed can be changed as follows:

Configuration options

`default`	set card to normal defaults.
`[no]nic`	[un]set nic mode.
`[no]eql`	[un]set equipment loopback. This is for testing only.
`(en|dis)ableA`	enable or disable port A for capture.
`(en|dis)ableB`	enable or disable port B for capture.
`[no]varlen`	dis/enable variable length capture. Otherwise record length padded to slen
`slen=X`	capture packets of X bytes long.
`long=X`	allow packets of X bytes long.
`[no]align64`	Generate records with 64-bit alignment [default 32-bit]
`mem=X:Y`	configure memory allocated to streams 0, 1,…..
`rxonly`	Assign all buffer memory to receive streams.
`txonly`	Assign all buffer memory to transmit streams.
`rxtx`	Assign buffer memory to transmit and receive streams.

Auto-negotiate

The DAG 4.3GE can operate in one of two modes, nic and nonic.

The nic mode assumes that the card is connected directly to a Gigabit Ethernet switch or card with a full-duplex cable, and the DAG will perform Gigabit Ethernet auto-negotiation.

The nonic mode is intended for use with optical fibre splitters. The receive socket of the DAG port is connected to the output of an optical splitter that is inserted into a network link between two other devices, and the transmit socket of the DAG is unconnected.

In this mode, Gigabit Ethernet auto-negotiation is not performed. One splitter on each DAG receive port can then be used to monitor each direction of a full-duplex Gigabit Ethernet link.

Ports

To affect one port, commands are applied to both ports by default. To affect only one port, use the `-a or -b options`. To disable a port for capturing, use the `disablea` and `disableb` commands.

4.4 DAG 4.3GE Card Capture Session

Description A successful DAG 4.3GE card capture session is accomplished by checking receiver ports optical signal levels and checking the card has correctly detected the link. This is followed by configuring DAG for normal use.

Procedure Follow these steps to troubleshoot DAG 4.3GE card configuration.

Step 1. Check Receiver Ports Optical Signal Levels.

The card supports 850 nanometer multimode fibre attachments with optical signal strength between 0 dBm and -22 dBm.

If in doubt, check card receiver ports light levels are correct using an optical power meter.

The card receiver ports are the lower of each dual-LC-style connectors, the closest to the PCI-X slot.

Cover unused ports with LC-style plugs to prevent dust and mechanical hazards from damaging optics.

Step 2. Check FPGA Image Loaded.

Before configuring the card, ensure the most recent FPGA image is loaded on the card.

```
dag@endace:~$ dagrom -rvp -d dag0 -f

dag@endace:~$ dagfour -d dag0
linkA   nonic noeql norxpkts notxpkts crc long=1518 enablea promisc
linkB   nonic noeql norxpkts notxpkts crc long=1518 enableb promisc
terf    terf_strip32
packet  varlen slen=48 noalign64
packetA drop=0
packetB drop=0
pcix    133MHz 64-bit buf=128MiB rxstreams=1 txstreams=1 mem=0:0
Firmware: edag43epci_terf_pci_terf_v2_9 2v1000ff896 2005/10/18
17:40:03
Card Serial: 3001658
MAC Address A: 00:0e:a7:00:52:6a
MAC Address B: 00:0e:a7:00:52:6b
```

Procedure (continued)

Step 3. Configure DAG for Normal Use.

The `dagfour default` command is always used:

```
dag@endace:~$ dagfour -d dag0 default
linkA   nonic noeql rxpkts txpkts crc long=1518 enablea promisc
linkB   nonic noeql rxpkts txpkts crc long=1518 enableb promisc
terf    terf_strip32
packet  varlen slen=2048 align64
packetA drop=0
packetB drop=0
pcix    133MHz 64-bit buf=128MiB rxstreams=1 txstreams=1 mem=112:16
Firmware: edag43epci_terf_pci_terf_v2_9 2v1000ff896 2005/10/18 17:40:03
Card Serial: 3001658
MAC Address A: 00:00:00:00:00:00
MAC Address B: 00:00:00:00:00:00
```

The `default` command always sets the DAG 4.3GE to nonic and noeql mode. For Ethernet link auto-negotiation use `default nic`.

Step 4. Check Card is Locked to Data Stream.

Configure card according to local settings.

Check through the physical layer statistics that the card is locked to the data stream.

4.5 Inspect Interface Statistics

Description Once the card has been configured, the interface statistics are inspected to check the card is locked to the data stream.

```
dag@endace:~$ dagfour -d dag0 -si
```

The tool displays a number of status bits that have occurred since last reading. The following example shows the interval is set to one second via the `-i` option.

`Sync`	A valid Gigabit Ethernet signal has been detected.
`Link`	This indicates that the Ethernet link is up.
`Auto`	In **nic** mode this indicates Ethernet autonegotiation has succeeded.
`RFlt`	The link peer is indicating a problem at the remote end.

Description (continued)

The following example shows the DAG 4.3GE in **`nic`** mode, with a Gigabit Ethernet router connected to port B via a full-duplex cable. Port A is unconnected.

```
dag@endace:~$ dagfour -d dag0 -si
```

```
   Port A:   Sync   Link   Auto   Rflt   Port B:   Sync   Link   Auto   RFlt
                0      0      0      0                1      0      1      0
                0      0      0      0                1      1      1      0
                0      0      0      0                1      1      1      0
                0      0      0      0                1      1      1      0
```

Extended statistics

Extended statistics are also available. The following example shows extended statistics from port B only for the above configuration.

```
dag@endace:~$  dagfour -d /dev/dag1 -bei
```

```
Port B:   Sync   Link   Auto   RFlt   Bad-Symb   CRC-Fail      Bytes      Frames
             1      1      1      0          0          0  15785982656   343173536
             1      1      1      0          0          0     68931230     1498505
             1      1      1      0          0          0     69130732     1502842
             1      1      1      0          0          0     69130870     1502845
```

NOTE: The first second has high values as the counters have accumulated their values over more than one second.

Statistic definitions

The extended statistic definitions include:

`Bad-Symb`	Bad Symbol Counter. Counts number of invalid Gigabit Ethernet Symbols received.
`CRC-Fail`	Local CRC Failure Counter. Number of Ethernet frames received that failed CRC check.
`Bytes`	Received Byte Counter
`Frames`	Received Frame Counter

4.6 Reporting Problems

Description

If there are unresolved problems with a DAG card or supplied software, contact Endace Technical Support via the email address support@endace.com. Supplying sufficient information in an email enables effective response.

Problem checklist

The exact information available to users for trouble, cause and correction analysis may be limited by nature of the problem. The following items assist a quick problem resolution:

Ref	Item
1.	DAG card[s] model and serial number.
2.	Host PC type and configuration.
3.	Host PC operating system version.
4.	DAG software version package in use.
5.	Any compiler errors or warnings when building DAG driver or tools.
6.	For Linux and FreeBSD messages generated when DAG device driver is loaded. These can be collected from command `dmesg` or from log file `/var/log/syslog`.
7.	Output of `daginf -v`.
8.	Firmware versions from `dagrom -x`.
9.	Physical layer status reported by: `dagfour`
10.	Network link statistics reported by: `dagfour -si`
11.	Network link configuration from the router where available.
12.	Contents of any scripts in use.
13.	Complete output of session where error occurred including any error messages from DAG tools. The `typescript` Unix utility may be useful for recording this information.
14.	A small section of a captured packet trace illustrating the problem.

Chapter 5: Running Data Capture Software

Introduction For a typical measurement session, ensure the driver is loaded, the firmware has been downloaded, and the card has been configured.

In this chapter This chapter covers the following sections of information.

- Starting Capture Session
- High Load Performance
- DAG 4.3GE Card Packet Transmission Capabilities

5.1 Starting Capture Session

Description The various tools used for data capture are in the `tools` sub-directory.

For a typical measurement session, ensure the driver is loaded, the firmware has been downloaded, and the card is configured.

The integrity of the card's physical layer is then set and checked.

Process Starting a data capture session is described in the following process.

Process	Description
Slen parameter default setting.	Slen parameter is set by default to 48 and long is set to 1518 in `dagfour`. If only part of a packet is required, such as for IP header capture, the value of `slen` can be changed using `dagfour`. `tools/dagfour slen=128 varlen`

Process (continued)

Process	Description
Setting capture session parameters	Parameters are set with `dagfour`. The card can operate in two modes, variable length capture (`varlen`), and fixed length capture (`novarlen`). In variable length capture mode, a maximum capture size is set with `slen=N` bytes. This figure should be in the range 32 to 10240 and is rounded down to the nearest multiple of 4. Packets longer than slen are truncated. Packets shorter than slen will produce shorter records, saving bandwidth and storage space. Full packet capture for example: `tools/dagfour -d dag0 varlen slen=1536`
Capturing non-standard ethernet frames.	For frames larger than 1500 Bytes in size, known as Jumbo frames, the value of 'long' is increased. For example, for full payload Jumbo frame capture: `tools/dagfour -d dag0 varlen slen=9600 long=9600`
Setting fixed length mode.	In fixed length mode, packets longer than the selected slen are truncated to slen. Packets shorter than slen produce records padded out to slen length. Large slen values in fixed length mode should be used because short packets arriving produce large padded records, wasting bandwidth and storage space. An example, for fixed length 64-byte records, choose slen=44 (64 – ERF header size of 16 – alignment padding 4) is: `tools/dagfour -d dag0 novarlen slen=44`

Process (continued)

Process	Description
Disabling individual ports.	Each direction [A and B] can be individually enabled and disabled for capture using dagfour. `tools/dagfour -d dag0 disableb`

Starting a capture session.	Once the capture parameters are configured, a capture session is started by:
	
	`tools/dagsnap -v -o tracefile`
	
	Option `-v` provides user information during capture; it can be omitted for automated trace runs.
	
	If the `-o tracefile` parameter is not specified the tool writes to stdout, which can be used to pipeline `dagsnap` with other tools from `dagtools` package.
	
	By default `dagsnap` runs forever. `dagsnap` can be stopped with a signal:
	
	`killall dagsnap`
	
	`dagsnap` can also be configured to run for a fixed number of seconds and then exit using the `-s flag`.

5.2 High Load Performance

Description

As the DAG card captures packets from the network link, it writes a record for each packet into a large buffer in the host PC's main memory.

Avoiding packet loss

In order to avoid packet loss, the user application reading the record, such as `dagsnap`, must be able to read records out of the buffer faster than they arrive. Otherwise the buffer eventually fills, and packet records are lost.

For Linux and FreeBSD, when the PC buffer becomes full, the message:

```
kernel: dagN: pbm safety net reached
```

is displayed on the PC screen, and printed to log `/var/log/messages`.

The "Data capture" LED also goes out. This may be visibly indicated as flashing or flickering.

Detecting packet losses

Until some data is read out of the buffer to free some space, any arriving packets subsequently are discarded by the DAG card.

Any loss can be detected in-band by observing the Loss Counter `lctr` field of the Extensible Record Format [ERF]. The Endace ERF is detailed in Chapter 7 of this document.

Increasing buffer size

The host PC buffer can be increased to deal with bursts of high traffic load on the network link.

By default the `dagmem` driver reserves 32MB of memory per DAG card in the system. Capture at OC-12/STM-4 (622Mbps) rates and above may require a larger buffer.

128MB or more is suggested for Linux/FreeBSD.

For the DAG 4.3GE card Windows operating system the upper limit is 128MB.

In Debian Linux the amount of memory reserved is changed by editing the file `/etc/modules`.

```
# For DAG 3.x, default 32MB/card
dagmem
#
# For DAG 4.x or 6.x, use more memory per card, E.G.
# dagmem dsize=128m
```

The option `dsize` sets the amount of memory used per DAG card in the system.

The value of `dsize` multiplied by the number of DAG cards must be less than the amount of physical memory installed, and less than 890MB.

5.3 DAG 4.3GE Card Packet Transmission Capabilities

Description The firmware included with the DAG 4.3GE card allows the DAG to transmit as well as receive packets, however the DAG does not appear as a network interface to the operating system.

In this chapter This chapter covers the following sections of information.

- DAG Packet Transmission
- Inline Forwarding

5.3.1 DAG Packet Transmission

Process The following information describes the DAG capabilities of the DAG firmware for the transmission and receiving of packets.

Process	Description
Explicit packets transmission.	The DAG will not respond to ARP, ping, or router discovery protocols. It will only transmit packets explicitly provided by the user. This capability allows the DAG card to be used as a simple traffic load generator. The DAG can also be used to retransmit previously recorded packet traces. The packet trace will be transmitted at 100% line rate, the packet timing of the original trace file is not reproduced.
Packet transmission utility	The `dagflood` utility can transmit ERF format packet traces. The ERF trace file to be transmitted must contain only ERF records of the type matching the current link configuration. The ERF records to be transmitted must all have a length which is a multiple of 64-bits. When capturing a packet trace for later transmission, you can set 64-bit alignment using the `dagfour align64` command.

Process,continued

Process	Description

Convert trace files.	It is also possible to convert trace files that have been captured without the `align64` option. This can be done with the command:
	`dagconvert -v -i in.erf -o out.erf -A8`
	If uncertain that a trace file is 64-bit aligned for transmission with `dagflood`, the file can be tested with `dagbits`:
	`dagbits -vvc align64 -f tracefile.erf`
	If a captured trace file is not available, the `daggen` program is capable of generating trace files containing simple traffic patterns. This allows the DAG card to be used as a test traffic generator.

Capture received traffic while transmitting.	It is possible to capture received traffic while transmitting. Capture programs such as `dagsnap`, `dagconvert`, and `dagbits` can be used while dagflood is sending packets. Use of 133MHz PCI-X is recommended to ensure adequate bandwidth is available for simultaneous receive and transmit operation.

Process (continued)

Process	Description
Configuring DAG card for transmission.	To configure a DAG card for transmission, some memory must be allocated to a transmit stream. In the `dagfour` output, `buf=nMB` indicates that `n` megabytes of memory has been allocated to this DAG card in total. This memory can be split between the available receive and transmit stream buffers. The memory allocation is displayed with `mem=X:Y`, where X is the amount of memory allocated to receive stream 0 in MB, and Y is the amount of memory allocated to transmit stream 1 in MB. By default the memory is evenly split between the receive streams, the transmit streams have no memory allocated. If the card is to be used only for transmit, the `dagfour txonly` option can be used to recover the receive buffer memory and assign all the memory to transmit. If the card is to be used for both transmitting and receiving, the `rxtx` option can be used. This allocates 16MB of memory to each transmit stream, and divides the remaining memory between the receive streams. Alternatively the memory allocation can be directly set with `mem=X:Y` option. The stream buffer memory allocation can only be changed when no packet capture or transmission programs are running.

5.3.2 Inline Forwarding

Description The DAG 4.3GE card can be used as an 'inline' device to receive, inspect, filter and forward packets between Port A and Port B.

Process The following information describes the DAG 4.3GE card inline forwarding process.

Process	Description
Inline transmission.	This operation can be performed at 100% line rate in both directions simultaneously. A PCI-X 133MHz slot is required for full performance and the performance may be limited by the host PC CPU and memory performance.
The 'dagfwddemo' program.	The 'dagfwddemo' program is provided as a demonstration of how this can be achieved. This program forwards packets bidirectionally, applying a user supplied BPF filter to each packet with the host CPU. Packets which match the filter are forwarded, while packets that do not match are dropped.
Modification of packets.	Modification of packets during inspection is also possible. The modifications should not change the length of the packet, and the user is responsible for re-computing checksums as needed. This is intended a demonstration of Inline Forwarding technology for use in Firewall or IDS/IPS applications. It is not suitable for use as a production Firewall.

Chapter 6: Synchronizing Clock Time

Description

The Endace DAG range of products come with sophisticated time synchronization capabilities, in order to provide high quality timestamps, optionally synchronized to an external time standard.

The system that provides the DAG synchronization capability is known as the DAG Universal Clock Kit (DUCK).

An independent clock in each DAG card runs from the PC clock. A card's clock is initialised using the PC clock, and then free-runs using a crystal oscillator.

Each card's clock can vary relative to a PC clock, or other DAG cards.

DUCK configuration

The DUCK is configured to avoid time variance between sets of DAG cards or between DAG cards and coordinated universal time [UTC].

Accurate time reference can be obtained from an external clock by connecting to the DAG card using the synchronization connector, or the host PCs clock can be used in software as a reference source without additional hardware.

Each DAG card can also output a clock signal for use by other cards.

Common synchronization

The DAG card synchronization connector supports a Pulse-Per-Second (PPS) input signal, using RS-422 signalling levels.

Common synchronization sources include GPS or CDMA (Cellular telephone) time receivers.

Endace produces the TDS 2 Time Distribution Server modules and the TDS 6 units that enable multiple DAG cards to be connected to a single GPS or CDMA unit.

More information is on the Endace website, http://www.endace.com/accessories.htm, or the TDS 2/TDS 6 Units Installation Manual.

In this chapter

This chapter covers the following sections of information.

- Configurations Tool Usage
- Time Synchronization Configurations
- Synchronization Connector Pin-outs

6.1 Configurations Tool Usage

Description

The DUCK is very flexible, and can be used in several ways, with or without an external time reference source. It can accept synchronization from several input sources, and can also be made to drive its synchronization output from one of several sources.

Synchronization settings are controlled by the `dagclock` utility.

Example

```
dag@endace:~$ dagclock -h
Usage: dagclock [-hvVxk] [-d dag] [-K <timeout>] [-l
<threshold>] [option]

     -h  --help,--usage   this page
     -v  --verbose        increase verbosity
     -V  --version        display version information
     -x  --clearstats     clear clock statistics
     -k  --sync           wait for duck to sync before
                          exiting
     -d  dag              DAG device to use
     -K  timeout          sync timeout in seconds, default
                          60
     -l  threshold        health threshold in ns, default
                          596
  Option:
        default           RS422 in, none out
        none              None in, none out
        rs422in           RS422 input
        hostin            Host input (unused)
        overin            Internal input (synchronize to
                          host clock)
        auxin             Aux input (unused)
        rs422out          Output the rs422 input signal
        loop              Output the selected input
        hostout           Output from host (unused)
        overout           Internal output (master card)
        set               Set DAG clock to PC clock
        reset             Full clock reset.  Load time
                          from PC, set rs422in, none out
```

By default, all DAG cards listen for synchronization signals on their RS-422 port, and do not output any signal to their RS-422 port.

```
dag@endace:~$ dagclock -d dag0
muxin   rs422
muxout  none
status  Synchronized Threshold 596ns Failures 0 Resyncs 0
error   Freq -30ppb Phase -60ns Worst Freq 75ppb Worst
Phase 104ns
crystal Actual 100000028Hz Synthesized 67108864Hz
input   Total 3765 Bad 0 Singles Missed 5 Longest Sequence
Missed 1
start   Thu Apr 28 13:32:45 2005
host    Thu Apr 28 14:35:35 2005
dag     Thu Apr 28 14:35:35 2005
```

6.2 Time Synchronization Configurations

Description The DUCK is very flexible, and can be used in several ways, with or without an external time reference source.

The use includes a single card with no reference, two cards with no reference, and a card with reference.

In this section This section covers the following topics of information.

- Single Card no Reference Time Synchronization
- Two Cards no Reference Time Synchronization
- Card with Reference Time Synchronization

6.2.1 Single Card no Reference Time Synchronization

Description When a single card is used with no external reference, the card can be synchronized to the host PC's clock.

The clock in most PC's is not very accurate by itself, but the DUCK drifts smoothly at the same rate as the PC clock.

If a PC is running NTP to synchronize its own clock, then the DUCK clock is less smooth because the PC clock is adjusted in small jumps. However, overall the DUCK clock does not drift away from UTC.

The synchronization achieved in this case is not as accurate as when using an external reference source such as GPS.

The DUCK clock is synchronized to a PC clock by setting input synchronization selector to overflow:

```
dag@endace:~$ dagclock -d dag0 none overin
muxin   overin
muxout  none
status  Synchronized Threshold 11921ns Failures 0 Resyncs
0
error   Freq 1836ppb Phase 605ns Worst Freq 143377ppb
Worst Phase 88424ns
crystal Actual 49999347Hz Synthesized 16777216Hz
input   Total 87039 Bad 0 Singles Missed 0 Longest
Sequence Missed 0
start   Wed Apr 27 14:27:41 2005
host    Thu Apr 28 14:38:20 2005
dag     Thu Apr 28 14:38:20 2005
```

NOTE: `dagclock` should be run only after appropriate Xilinx images have been loaded. If the Xilinx images must be reloaded, the `dagclock` command must be rerun afterwards to restore the configuration.

6.2.2 Two Cards no Reference Time Synchronization

Description

When two DAG cards are used in a single host PC with no reference clock, the cards are to be synchronized in some way if timestamps between the two cards are to be compared. For example, if two cards monitor different directions of a single full-duplex link.

Synchronization between two DAG cards is achieved in two ways. One card can be a clock master for the second, or one can synchronize to the host and also act as a master for the second.

Synchronizing cards

If both cards are to be accurately synchronized, but not so for absolute time of packet time-stamps being correct, then one card is configured as the clock master for the other.

Locking cards together

Although the master card's clock will drift against UTC, the cards are locked together.

The cards are locked together by connecting the synchronization connector ports of both cards with a standard RJ-45 Ethernet cross-over cable.

Configure one of the cards as the master, the other defaults to being a slave.

```
dag@endace:~$ dagclock -d dag0 none overout
muxin   none
muxout  over
status  Not Synchronized Threshold 596ns Failures 0
Resyncs 0
error   Freq 0ppb Phase 0ns Worst Freq 0ppb Worst Phase
0ns
crystal Actual 100000000Hz Synthesized 67108864Hz
input   Total 0 Bad 0 Singles Missed 0 Longest Sequence
Missed 0
start   Thu Apr 28 14:48:34 2005
host    Thu Apr 28 14:48:34 2005
dag     No active input - Free running
```

The slave card configuration is not shown, the default configuration is sufficient.

Preventing time-stamps drift

To prevent the DAG card clocks time-stamps drifting against UTC, the master can be synchronized to the host PC's clock which in turn utilises NTP. This then provides a master signal to the slave card.

The cards are locked together by connecting the synchronization connector ports of both cards with a standard RJ-45 Ethernet cross-over cable.

Configure one card to synchronize to the PC clock and output a RS-422 synchronization signal to the second card.

```
dag@endace:~$ dagclock -d dag0 none overin overout
muxin   over
muxout  over
status  Synchronized Threshold 11921ns Failures 0 Resyncs
0
error   Freq -691ppb Phase -394ns Worst Freq 143377ppb
Worst Phase 88424ns
crystal Actual 49999354Hz Synthesized 16777216Hz
input   Total 87464 Bad 0 Singles Missed 0 Longest
Sequence Missed 0
start   Wed Apr 27 14:27:41 2005
host    Thu Apr 28 14:59:14 2005
dag     Thu Apr 28 14:59:14 2005
```

The slave card configuration is not shown, the default configuration is sufficient.

6.2.3 Card with Reference Time Synchronization

Description

The best timestamp accuracy occurs when DAG card is connected to an external clock reference, such as a GPS or CDMA time receiver.

Pulse signal from external sources

The DAG synchronization connector accepts a RS-422 Pulse Per Second [PPS] signal from external sources.

This is derived directly from a reference source, or distributed through the Endace TDS 2 [Time Distribution Server] module which allows two DAG cards to use a single receiver.

More cards can be accommodated by daisy-chaining TDS-6 expansion units to the TDS-2 unit, each providing outputs for an additional 6 DAG cards.

Using external reference source

To use an external clock reference source, the host PC's clock must be accurate to UTC to within one second. This is used to initialise the DUCK.

The external time reference allows high accuracy time synchronization.

When the time reference source is connected to the DAG synchronization connector, the card automatically synchronizes to a valid signal.

```
dag@endace:~$ dagclock -d dag0
muxin rs422
muxout none
status Synchronized Threshold 596ns Failures 0 Resyncs 0
error Freq 30ppb Phase -15ns Worst Freq 2092838ppb Worst
Phase 33473626ns
crystal Actual 100000023Hz Synthesized 67108864Hz
input Total 225 Bad 0 Singles Missed 1 Longest Sequence
Missed 1
start Thu Apr 28 14:55:20 2005
host Thu Apr 28 14:59:06 2005
dag Thu Apr 28 14:59:06 2005
```

Connecting time distribution server

The TDS 2 module connects to any DAG card with a standard RJ-45 Ethernet cable and can be placed some distance from a DAG card.

Existing RJ-45 building cabling infrastructure can be used to cable synchronization ports.

CAUTION: Never connect DAG and/or the TDS 2 module to active Ethernet or telephone equipment.

Testing signal

For Linux and FreeBSD, when a synchronization source is connected the driver outputs some messages to the console log file `/var/log/messages`.

The `dagpps` tool is used to test a signal is being received correctly and is of correct polarity. To perform the test, run:

```
dagpps -d dag0.
```

The tool measures input state many times over several seconds, displaying polarity and length of input pulse.

Some DAG cards have LED indicators for synchronization (PPS) signals.

6.3 Synchronization Connector Pin-outs

Description DAG cards have an 8-pin RJ45 connector with two bi-directional RS422 differential circuits, A and B. The PPS signal is carried on circuit A, and the serial packet is connected to the B circuit.

Pin assignments The 8-pin RJ45 connector pin assignments are:

1.	Out A+
2.	Out A-
3.	In A+
4.	In B+
5.	In B-
6.	In A-
7.	Out B+
8.	Out B-

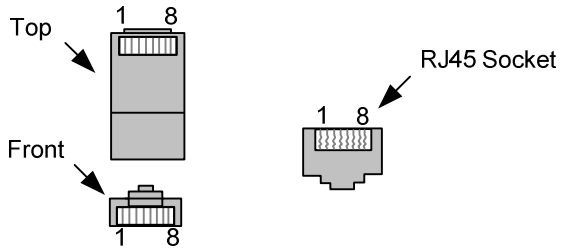

Figure 6-1. RJ45 Plug and Socket Connector Pin-outs.

Out-pin connections Normally the GPS input should be connected to the A channel input, pins 3 and 6. The DAG can also output a synchronization pulse; used when synchronizing two DAG's without a GPS input. Synchronization output is generated on the Out A channel, pins 1 and 2.

Ethernet crossover cable A standard Ethernet crossover cable can be used to connect the two cards.

TX_A+	1	3	RX_A+
TX_A-	2	6	RX-A-
RX_A+	3	1	TX_A+
RX_B+	4	7	TX_B+
RX_B-	5	8	TX_B-
RX_A-	6	2	TX_A-
TX_B+	7	4	RX_B+
TX_B-	8	5	RX_B-

Support For cables and further advice on using GPS and CDMA time receivers email support@endace.com.

Chapter 7: Data Formats Overview

In this chapter This chapter covers the following sections of information.

- Data Formats
- Timestamps

7.1 Data Formats

Description The DAG 4.3GE card uses the ERF Type 2 Ethernet Variable Length Record. Timestamps are in little-endian [Pentium native] byte order. All other fields are in big-endian [network] byte order. All payload data is captured as a byte stream, no byte re-ordering is applied.

timestamp		
timestamp		
type	flags	rlen
lctr		wlen
(rlen - 16) bytes of record		

Table 7-1. Generic Variable Length Record.

Data format The following is an overview of the data format used.

Data Format	Description
type:	This field contains an enumeration of the frame subtype. If the type is zero, then this is a legacy format. 0: TYPE_LEGACY 1: TYPE_HDLC_POS: PoS w/HDLC framing 2: TYPE_ETH: Ethernet 3: TYPE_ATM: ATM Cell 4: TYPE_AAL5: reassembled AAL5 frame 5: TYPE_MC_HDLC: Multi-channel HDLC frame 6: TYPE_MC_RAW: Multi-channel Raw link data 7: TYPE_MC_ATM: Multi-channel ATM Cell

Data Format	Description
flags:	This byte is divided into 2 parts, the interface identifier, and the capture offset. 1-0: capture interface 0-3 2: varying record lengths present 3: truncated record [insufficient buffer space] 4: rx error [link error] 5: 5: ds error [internal error] 7-6: reserved
Rlen: record length	Total length of the record transferred over PCI bus to storage.
Lctr: *loss counter*	A 16 bit counter, recording the number of packets lost since the previous record. Records can be lost between the DAG card and memory hole due to overloading on PCI bus. The counter starts at zero, and sticks at 0xffff.
Wlen: *wire length*	Packet length including some protocol overhead. The exact interpretation of this quantity depends on physical medium.
offset:	Number of bytes *not* captured from start of frame. Typically used to skip link layer headers when not required in order to save bandwidth and space. This field is currently not implemented, contents can be disregarded.

timestamp		
timestamp		
type:2	flags	rlen
lctr		wlen
offset	pad	rlen-18
bytes of frame		

Table 7-2. Type 2 Ethernet Variable Length Record.

The Ethernet frame begins immediately after the pad byte so that the layer 3 [IP] header is 32Bit-aligned.

7.2 Timestamps

Description

The ERF format incorporates a hardware generated timestamp of the packet's arrival.

The format of this timestamp is a single little-endian 64-bit fixed point number, representing seconds since midnight on the first of January 1970.

The high 32-bits contain the integer number of seconds, while the lower 32-bits contain the binary fraction of the second. This allows an ultimate resolution of 2^{-32} seconds, or approximately 233 picoseconds.

Another advantage of the ERF timestamp format is that a difference between two timestamps can be found with a single 64-bit subtraction. It is not necessary to check for overflows between the two halves of the structure as is needed when comparing Unix time structures, which are also available to Windows users in the Winsock library.

Different DAG cards have different actual resolutions. This is accommodated by the lowermost bits that are not active being set to zero. In this way the interpretation of the timestamp does not need to change when higher resolution clock hardware is available.

Example code

Here is some example code showing how a 64-bit ERF timestamp (erfts) can be converted into a struct timeval representation (tv).

```
unsigned long long lts;
struct timeval tv;

 lts = erfts;
 tv.tv_sec = lts >> 32;
 lts = ((lts & 0xffffffffULL) * 1000 * 1000);
 lts += (lts & 0x80000000ULL) << 1;      /* rounding */
 tv.tv_usec = lts >> 32;
 if(tv.tv_usec >= 1000000) {
    tv.tv_usec -= 1000000;
    tv.tv_sec += 1;
      }
```