DAG 4.5G2/G4/GF Card
User Guide

EDM01-18v3

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

Overview

The installation of the Endace DAG 4.5G2/G4/GF 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.

This document, DAG 4.5G2/G4/GF Card User Manual is available on the installation CD.

Purpose

Description

The purpose of this DAG 4.5G2/G4/GF Card User Manual is to describe:

- Installing the DAG 4.5G2/G4/GF cards
- Setting optical power
- Confidence testing
- Running data capture software
- Synchronizing clock time
- Data formats overview

Pre-requisites

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 4.5G2/G4/GF Card Product Description

Card Description

The DAG 4.5G2/G4/GF card is a PCI-X bus card designed for cell and packet capture and generation on Ethernet networks and is shown below:

The DAG 4.5G2/G4/GF 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 4.5G2/G4/GF is capable of transmitting packets at 100% line rate on all ports (G2 has two ports and G4 has four ports) while simultaneously receiving packets at 100% line rate on all ports. The GF (failsafe) version of the card has 2 ports only.

Card Architecture

Description

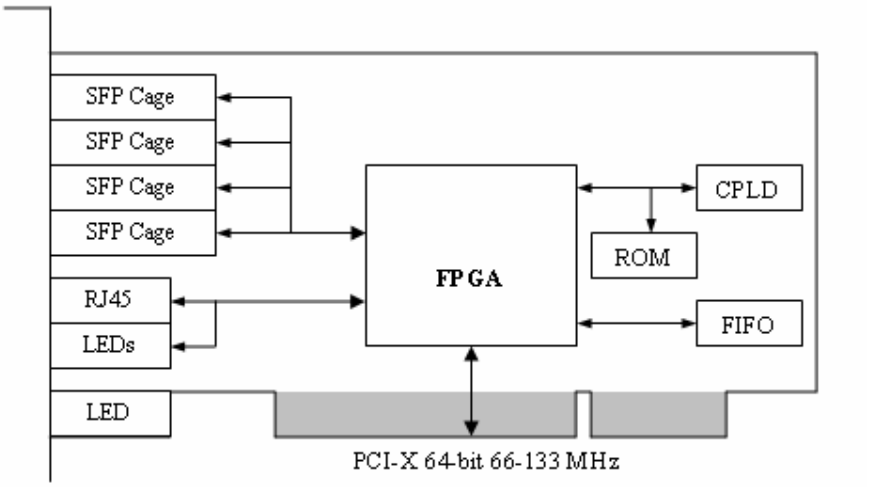
Serial Ethernet optical network data received by two 1000baseSX optical interfaces flow directly into the Field-Programmable Gate Array [FPGA].

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.

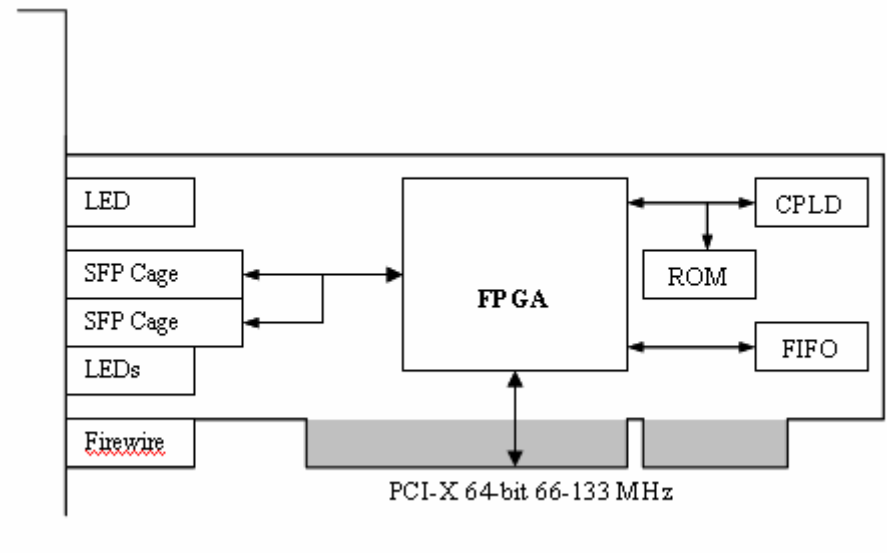
DAG 4.5G4

The diagram below shows the DAG 4.5G4 major components and processing flow:



DAG 4.5G2/GF

The diagram below shows the DAG 4.5G2 Card major components and process flow.



Note: The DAG 4.5GF is the same as the DAG 4.5G2 except for the failsafe relays.

1.4 DAG 4.5G2/G4/GF Card System Requirements

System requirements

General

The DAG 4.5G2/G4/GF 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
- PC, at least Intel Xeon 1.8GHz or faster
- 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.

Other Systems

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 4.5G2/G4/GF 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 4.5G2/G4/GF 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.

DAG Device Driver

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.

Inserting the Card

Inserting the DAG 4.5G2/G4/GF card into a PC involves accessing the PCI-X bus slot, fitting the card, and secure the bus slot screw. Follow the steps below to insert the DAG 3.7D card in the computer.

- Power computer down.
- Remove PCI bus slot screw and cover
- Insert DAG 3.7D card into PCI bus slot.
- Ensure free end fits securely into a card-end bracket that supports the card weight.
- Secure the card with the cover screw.
- Power the computer up

Port Connectors

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 while the bottom connectors for each pair are used for the received signals.

An 8-pin RJ45 socket in the 4.5G4 and a 4-pin Firewire socket in the 4.5G2 are used for time synchronization. These sockets should never be connected to anything other than Endace approved time synchronization sources.

Pluggable Optical Transceivers

All DAG 4.5G2/G4/GF cards are available with pluggable optics as shown below:

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 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 on the Pluggable Optical Transceiver visit http://www.endace.com/dagPluggable.htm .

Setting Optical Power

Overview

The optical power range depends on the particular device fitted on the DAG 4.5G2/G4/GF card. The card is shipped fitted with two 1000baseSX FTRJ 8519F 850nm multi-mode short range optics modules by default.

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.

The DAG 4.5G2/G4/GF card optics power module configuration.is shown below:

Part	Fibre	Data Rate	Max Pwr (dBm)	Min Pwr (dBm)	Nom Pwr (dBm)
FTR8519F	MMF	1000	0	-22	-14

Power Input

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 of 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.

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.

Splitter Losses

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

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

Note: A single mode fibre connected to a multi-mode input has minimal extra loss.
A multi-mode fibre connected to a single mode input creates large and unpredictable loss.

Chapter 3: Confidence Testing

Introduction

The confidence testing is a process to determine the DAG 4.5G2/G4/GF card is functioning correctly.

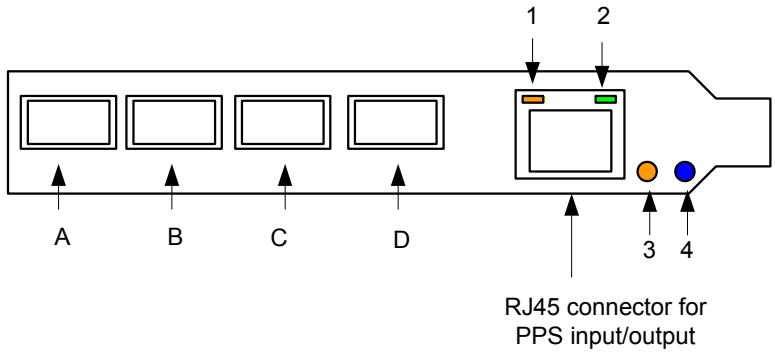
The process also involves a card capture session, and demonstrates configuration in the style of 'What You See You Can Change', WYSYCC.

Interface statistics are also inspected during this process.

LED Status

DAG4.5G4

The DAG 4.5G4 has 4 status LEDs, one coloured blue, two orange and , one green shown below:



When a DAG 4.5G24 series card is powered up the LED 1 should always come on.

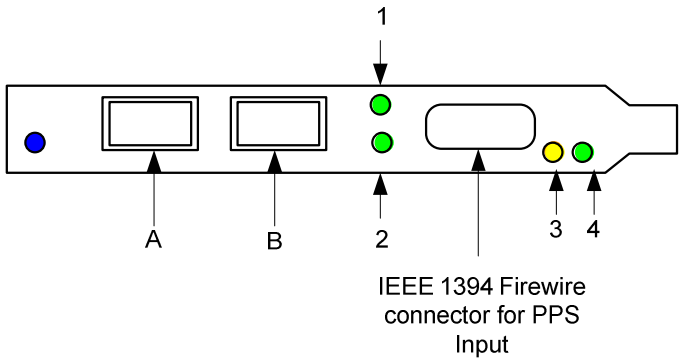
The LED definitions are shown below:

LED	Description
1	PPS Out: Pulse Per Second Out – indicates card is sending a clock synchronization signal.
2	PPS In: Pulse Per Second In – indicates card is receiving an external clock synchronization signal.
3	Data capture in progress
4	FPGA successfully programmed

LED Status

DAG4.5G2/GF

The DAG 4.5G2 has 5 status LEDs, one coloured blue, three green, and one amber as shown below:



The LED definitions are shown below:

LED	Description
1	Indicates Port A has link
2	Indicates Port has link
3	Data capture in progress
4	PPS In: Pulse Per Second In – indicates card is receiving an external clock synchronization signal
5	FPGA successfully programmed

Card Configuration Options

Configuration in WYSYCC is the 'What You See You Can Change' style. Running the command '`dagconfig`' alone shows the current configuration. Each of the items displayed can be changed as follows:

default	Set card to normal defaults.
[no]nic	[un]set nic mode.
[no]eql	[un]set equipment loopback. This is for testing only.
[no]rktiopwr	Power-on[off] high speed serial links to SFP modules
[no]laser	Enable/Disable transmit laser in the SFP modules
(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.
[no]align64	Generate records with 64-bit alignment [default 32-bit]
mem=X:Y	Configure memory allocated to streams 0, 1,…..

Auto-negotiation

The DAG 4.5G2/G4/GF 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

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.

Card Capture Session

Overview

A successful DAG 4.5G2/G4/GF 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.

For the GF version you need to ensure that you engage the failsafe relays.

Follow these steps shown below to troubleshoot DAG 4.5G2/G4/GF card configuration.

Engaging Failsafe Relays (DAG4.5 GF only)

The dogwatch command is used to activate the failsafe relays and connect the line to the physical layer interface on the card. To do this run the following command:

```
dagwatchdog –p –d N
```

where "`N`" is the number of the DAG Card on which the relays you want to engage are located

Check Receiver Port 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.

Check FPGA Image Loaded

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

```
dag@endace:~$ dagrom -d 0 -ryvp -f
<path>/dag45gepci_terf.bit


dag@endace:~$ dagconfig -d0

Card Information:

Firmware: edag45gepci_terf_v2_1 2vp30ff1152 2006/03/08
13:37:08 (user)

Serial: 3001981

MAC Address A: 00:00:00:00:00:00

MAC Address B: 00:00:00:00:00:00


linkA: nic noeql rktiopwr nolaser nosfppwr detect

linkB: nic noeql rktiopwr nolaser nosfppwr detect


GPP:

varlen slen=48 align64

linkA: drop_count=0 enablea

linkB: drop_count=0 enableb


PCI Burst Manager

133MHz buffer size=128 rx_streams=1 tx_streams=1
mem=0:0

TERF: terf_strip32
```

Configuration for Normal Use

The `dagconfig default` command is always used:

```
dag@endace:~$ dagconfig default
Card Information:
Firmware: edag45gepci_terf_v2_1 2vp30ff1152 2006/03/08
13:37:08 (user)
Serial: 30011948
MAC Address A: 00:00:00:00:00:00
MAC Address B: 00:00:00:00:00:00

linkA: nic noeql rktiopwr nolaser sfppwr detect
linkB: nic noeql rktiopwr nolaser sfppwr detect

GPP:
varlen slen=48 align64
linkA: drop_count=0 enablea
linkB: drop_count=0 enableb

PCI Burst Manager
133MHz buffer size=128 rx_streams=1 tx_streams=1
mem=112:16
TERF: terf_strip32
```

The `default` command always sets the DAG 4.5G2/G4/GF to nic and noeql mode. For Ethernet link auto-negotiation use `default laser`

The 4.5GF has relays for inline forwarding applications to reconnect the two ports in case of power failure. When the relays are in this state, the ports are not connected to the physical layer devices on the card. To use the card in such case the relays must be engaged run:

```
dagwatchdog -p -d N
```

Where `N` is the number of the DAG card to engage the relays on.

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.

Interface Statistics

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

```
dag@endace:~$ dagconfig -d0 -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.

`LOS`	NO valid Gigabit Ethernet signal detected.
`LOF`	Internal framer has lost synchronisation.
`RFault`	The link peer is indicating a problem at the remote end.
`Plink`	The link peer is indicating a valid link.
`Link`	In **nic** mode this indicates a fully auto-negotiated link

The following example shows a DAG 4.5G2 in **`nic`** mode, with a Gigabit Ethernet router connected to port B via a full-duplex cable. Port A is unconnected.

```
dag@endace:~$ dagconfig -d0 -si
```

Port	Link	PLink	RFault	LOF	LOS
A	0	0	0	1	1
B	1	1	0	0	0

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

```
dag@endace:~$  dagconfig -d0 -2ei
```

Port	TxFault	MiniMacLostSync	Remote Error	LOF	LOS	Peer Link	Link
B	0	0	0	0	0	1	1
B	0	0	0	0	0	1	1
B	0	0	0	0	0	1	1
B	0	0	0	0	0	1	1

The extended status definitions include:

`TxFault`	SFP module status during transmission.
`MiniMacLostSync`	Internal framer has lost synchronisation with peer.

Reporting Problems

If you have problems with a DAG card or Endace supplied software which you are unable to resolve, please contact Endace Customer Support at support@endace.com.

Supplying as much information as possible enables Endace Customer Support to be more effective in their response to you. The exact information available to you for troubleshooting and analysis may be limited by nature of the problem. However the following items will assist a quick resolution:

- DAG card[s] model and serial number.
- Host PC type and configuration.
- Host PC operating system version
- DAG software version package in use
- Any compiler errors or warnings when building DAG driver or tools
- 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`.
- Output of `daginf`
- Firmware versions from `dagrom -x`.
- Physical layer status reported by: `dagthree`
- Network link statistics reported by: `dagthree -si`
- Network link configuration from the router where available.
- Contents of any scripts in use.
- 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.
- A small section of captured packet trace illustrating the problem.

Chapter 4: Running Data Capture Software

Overview

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

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.

Starting a Capture Session

Follow the steps described below to start a data capture session:

Setting the Slen Parameter Default

- Slen parameter is set by default to 48 bytes.
- If only part of a packet is required, such as for IP header capture, the value of `slen` can be changed using `dagconfig`.

```
tools/dagf
```

Setting Session Parameters

- Parameters are set with `dagconfig`.
- 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 8.
- 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/dagconfig -d0 varlen slen=1536
```

Capturing Non-Ethernet Frames

For frames larger than 1500 Bytes in size, known as Jumbo frames, the value of slen is increased. For example, for full payload Jumbo frame capture:

```
tools/dagconfig -d0 varlen slen=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=40 (64 – ERF header size of 18 – alignment padding 6) is:

```
tools/dagconfig -d0 novarlen slen=40
```

Disabling Individual Ports

Each direction [A and B] can be individually enabled and disabled for capture using dagfour.

```
tools/dagconfig -d0 disableb
```

Beginning the 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` will run indefinitely but can be stopped using *ctrl+C* You can also configure `dagsnap` to run for a fixed number of seconds and then exit using the `-s flag`.

High Load Performance

Overview

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

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 6 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 4.5G2/G4/GF 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.

Packet Transmission

Overview

The firmware included with the DAG 4.5G2/G4/GF 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.

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

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 `dagconfig align64` command

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 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.

Configuring Card for Transmission

To configure a DAG card for transmission, some memory must be allocated to a transmit stream.

In the `dagconfig` output, `buffer size=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 more memory is given to the receive streams compared to the transmit streams.

The stream buffer memory allocation can only be changed when no packet capture or transmission programs are running.

In-Line Forwarding

Overview

The DAG 4.5G2/G4/GF card can be used as an 'inline' device to receive, inspect, filter and forward packets between Port A and Port B.

The following information describes the DAG 4.5G2/G4/GF card inline forwarding process.

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.

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 5
Synchronizing Clock Time

Introduction

Overview

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

The system that provides the DAG synchronisation 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 synchronisation 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 synchronisation connector supports a Pulse-Per-Second (PPS) input signal, using RS-422 signalling levels.

Common synchronisation 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 available on the Endace website at http://www.endace.com/accessories.htm , or the *TDS 2/TDS 6 Units Installation Manual*.

Configuration Tools

Description

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

Synchronisation 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 (synchronise 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 synchronisation signals on their RS-422 port, and do not output any signal to their RS-422 port

Configuration Tools (cont.)

```
dag@endace:~$ dagclock -d dag0
muxin   rs422
muxout  none
status  Synchronised 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
```

Single Card No Reference

Overview

When a single card is used with no external reference, the card can be synchronised to the host PC clock. Most PC clocks are not very accurate by themselves, but the DUCK drifts smoothly at the same rate as the PC clock.

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

The synchronisation achieved 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  Synchronised 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 are reloaded, the `dagclock` command must be rerun afterwards to restore the configuration.

Two Cards No Reference

Overview

When two DAG cards are used in a single host PC with no reference clock, the cards must 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 synchronise to the host and also act as a master for the second.

Synchronising Cards

If both cards are to be accurately synchronised, 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 synchronisation 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 Synchronised 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 clock time-stamps drifting against UTC, the master can be synchronised 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 synchronisation 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  Synchronised 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.

Card with Reference

Overview

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

Pulse Signal from External Source

The DAG synchronisation 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 an 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 synchronisation.

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

dag@endace:~$ dagclock –d dag0

muxin rs422

muxout none

status Synchronised 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 the 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 synchronisation ports.

The TDS-2 and the DAG card synchronisation port should never be connected to Ethernet or telephone equipment.

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

Testing the Signal

For Linux and FreeBSD, when a synchronisation 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 synchronisation (PPS) signals.

Connector Pin-outs

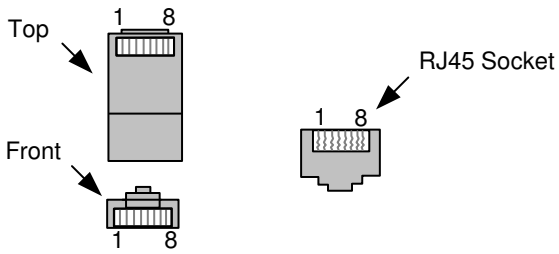
Overview

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 and plugs and sockets are shown below:

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



Out-pin Connections

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

Ethernet Crossover Table

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-

Chapter 6:
Data Formats Overview

Overview

The DAG 4.5G2/G4/GF 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.

Generic Header

All ERF records share some common fields. 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.

The generic ERF header is shown below.

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

•

timestamp	The time of arrival of the cell, an ERF 64-bit timestamp.
type	2: TYPE_ETH (Ethernet) 16: TYPE_DSM_COLOR_ETH (DSM Ethernet)
flags	This byte is divided into 2 parts, the interface identifier and the capture offset.
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 the PCI bus. The counter starts at 0 and sticks as 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 form start of frame. Typically used to skip link layer headers when not required to save bandwidth and space. This is not currently implemented, contents can be disregarded.

Ethernet Record

The Ethernet frame begins immediately after the pad byte so that the layer 3 [IP] header is 32Bit-aligned. The `Type 2` record is described below:

BYTE 3	BYTE 2	BYTE 1	BYTE 0
timestamp			
timestamp			
type 2	flags	rlen	
lctr		wlen	
offset	pad	rren-18	
(rlen - 20) bytes of record			

DSM Ethernet Record

The Type 16 record is shown below:

BYTE 3	BYTE 2	BYTE 1	BYTE 0
timestamp			
timestamp			
type 2	flags	rlen	
color		wlen	
offset	pad	rren-18	
(rlen - 20) bytes of record			

Timestamps

Overview

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

Below 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;
       }
```