endace
accelerated

DAG 3.7G Card User Guide

EDM01-07

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

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

Table of Contents (cont.)

Chapter 1: Introduction

Overview

The Endace DAG 3.7G series consist of two PCI-bus card types, DAG 3.7GF and the DAG 3.7GP.

The installation of an Endace DAG 3.7G series 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.

Purpose of this User Guide

The purpose of this User Guide is to provide you with an understanding of the DAG card architecture and functionality and to guide you through the following:

- Installing the Card and associated software and firmware
- Configuring the card for your specific network requirements
- Running a data capture session
- Synchronising clock time
- Data formats

You can also find additional information relating to functions and features of the DAG 3.7G card in the following documents which are available from the Support section of the Endace website at www.endace.com:

- EDM04-08 Configuration and Status API Programming Guide,

This User Guide and the Linux and Window Guides are also available in PDF format on the Installation CD shipped with your DAG 3.7G card.

System Requirements

General

The minimum system requirements for the DAG 3.7G card are :

- PC, at least Pentium II 400 MHz, Intel 440BX, GX or newer chip set
- 256 MB RAM
- At least one free 3.3V 32 or 64 bit PCI slot
- 30MB free disk space for software distribution

Note: A 64-bit PCI slot is recommended in order to maximize performance.

Operating System

This User Guide assumes you are installing the DAG card in a PC which already has an operating system installed.

However for convenience, a copy of Debian Linux 3.1 (Sarge) is provided as a bootable ISO image on the CDs that is shipped with the DAG card.

To install either the Linux/FreeBSD or Windows operating system please refer to the following documents which are also included on the CD that is shipped with the DAG card.

- EDM04-01 Linux FreeBSD Software Installation Guide
- EDM 04-02 Windows Software Installation Guide

Other Systems

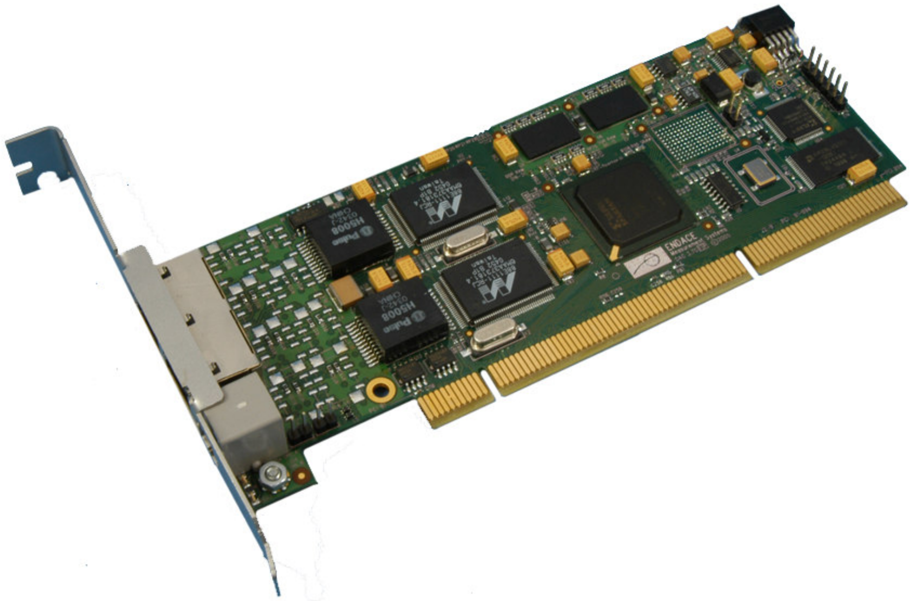
For advice on using an operating system that is substantially different from either of those specified above, please contact Endace Customer Support at support@endace.com

Card Description

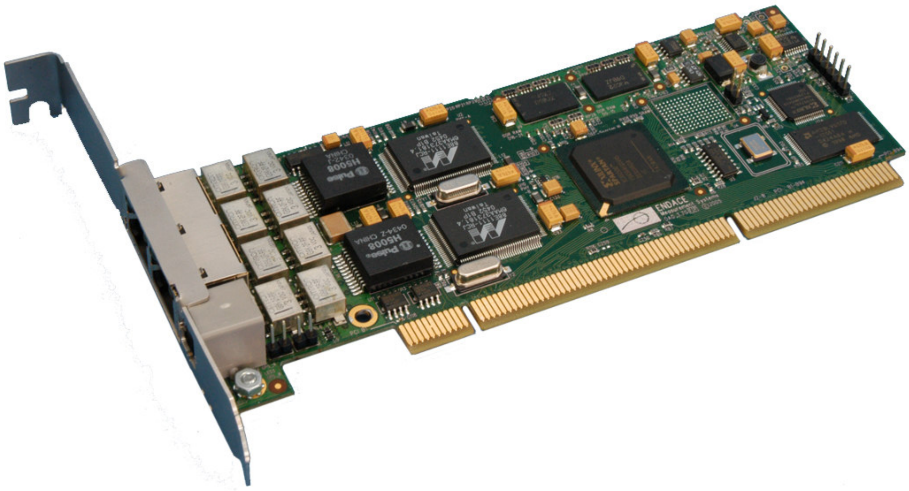
The DAG 3.7GF has failsafe relays to connect the two ports on the card in event of a power failure. This failsafe feature is intended for use in inline forwarding applications. The DAG 3.7GP does not have the failsafe feature.

The DAG Ethernet ports will operate in half duplex or full duplex modes. The DAG 3.7G series card by default finds the fastest link configuration possible with the peer device using Ethernet Autonegotiation.

The DAG 3.7GP card is shown below:



The DAG 3.7GF card is shown below:



Card Architecture

Overview

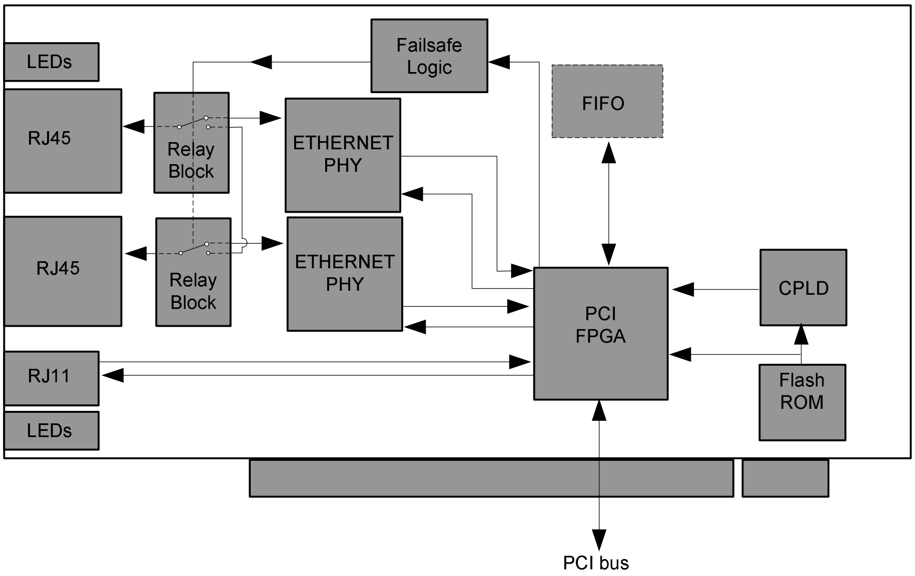
The DAG 3.7G series card is designed for packet capture and generation on Ethernet networks.

Ethernet data is received by a DAG 3.7G series card interfaces, and fed through framers into the Xilinx FPGA.

This FPGA contains an Ethernet processor and the DUCK timestamp engine.

Because of close association of the components, packets are time-stamped accurately. Time stamped packet records are stored by the FPGA, which interfaces to the PCI bus. All packet records are written to host PC memory during capture operations.

The following diagram shows the card's major components and the flow data:



NIC Functionality

The DAG 3.7G series card have two 10/100/1000 Mbps Copper Ethernet There default configuration is as if the DAG card was a NIC, and can be connected to a hub, switch or router port directly.

Each DAG 3.7G port can also be connected to a NIC card. The DAG 3.7G cards support automatic MDI/MDI-X switching, so can be connected to a NIC using either an Ethernet straight-through or cross-over cable. When using the failsafe feature of the DAG 3.7GF, there are some advantages to using a straight through cable rather than a cross-over one. The DAG card captures all packets received on each port, similar to a NIC in promiscuous mode.

Memory Holes

Memory hole configuration is dependant on the application requirements. For a receive-only configuration, two memory holes are available, on each port.

For packet forwarding applications, only one memory hole can be utilised.

Failsafe Relays

The DAG 3.7GF card failsafe relays are capable of either:

- Connecting the two ports together as a pass-through link
- Connecting both ports to the FPGA to enable data capture. This feature is not available on 3.7GP cards.

Chapter 2: Installation

Introduction

Note: Throughout this document the "*DAG 3.7G*" refers to both the DAG 3.7GF card and the DAG 3.7GP card.

The DAG 3.7G card can be installed in any free 32-bit or 64-bit Bus Mastering PCI slot.

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-bus.

The cards make very heavy use of PCI-bus data transfer resources. This is not usually a limitation as for most applications a maximum of two cards only can be used with reasonable application performance.

DAG Device Driver

The DAG device driver must be installed before you install the DAG card itself.

If you have not already completed this please follow the instructions in *EDM04-01 Linux FreeBSD Software Installation Guide* or E*DM 04-02 Windows Software Installation Guide* as appropriate, which are included on the CD shipped with the DAG card.

Inserting the DAG Card

To insert the DAG card in the PC follow the steps described below:

- Turn power to the computer OFF,
- Remove the PCI bus slot screw and cover,
- Insert DAG card into PCI bus slot ensuring that it is firmly seated in the slot,
- Check the free end of the card fits securely into the card-end bracket that supports the weight of the card,
- Secure the card with the bus slot screw,
- Turn power to the computer ON.

Connecting the Interfaces

There are two RJ45 connectors on the DAG 3.7G card, and a RJ11 connector.

The RJ45 connectors, furthest from PCI connector, are the network monitoring ports. These can be connected directly to Ethernet Hubs, Switches or Router ports with a standard Ethernet cable. The monitoring ports can also be connected directly to NIC cards using either ethernet cross-over or straight-through cables.

The RJ11 socket, near the PCI connector, is for the time synchronization input. This socket should never be connected to a telephone line.

Card Sensitivity

The DAG 3.7G card monitoring ports conform to the IEEE 802.3 standard for Ethernet.

The standard specifies a maximum cable length of 100 metres for 10Base-T, 100-BaseTX, and 1000Base-T operation over unshielded twisted pair CAT5E or better cable.

By default DAG 3.7G card automatically detects line speed of 10, 100, or 1000Mbps.

Light link status lights indicate the network is detected correctly.

Activity lights indicate network traffic.

Chapter 3: Configuring the Card

Introduction

Configuring the DAG card for data capture involves:

- Engaging failsafe relays,
- Interpreting card LED status,
- Starting a capture session,
- Inspecting statistics.

Engaging Failsafe Relays

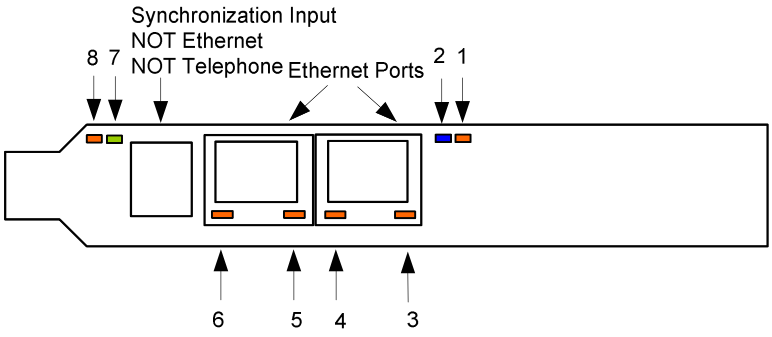
The 3.7GF has relays for inline forwarding applications to reconnect the two ports in case of power failure. When the relays are in this state, the ports are not connected to the physical layer devices on the card. To use the card in such case the relays must be engaged. Run:

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

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

LEDs and Inputs

Before you begin to configure the DAG card it is important to understand the function of the various LEDs associated with the card, as well as the sockets on the PCI bracket.



The LED functions are described next:

LED	Description
1	Burst manager run; Indicates card is capturing packets and transferring them to the host
2	FPGA successfully programmed.
3	Port A Activity
4	Port A Link up
5	Port B Activity
6	Port B Link up
7	PPS In: Pulse Per Second In; Indicates card is receiving an external clock synchronization signal. Inactive when PPS cable not plugged in.
8	PPS Out: Pulse Per Second Out; Blinking indicates the card is sending a clock synchronization signal.

Configuration Utility

The `dagthree` utility supports configuration and reading of card status and physical layer interface statistics for the DAG 3.x series of cards. In a troubleshooting configuration options, `dagthree -si` will display the operational status of the physical and framing layers, updated once per second.

More details about the meaning of the various parameters/options are supplied through the help page (`dagthree -h`) as well as via the manual page.

Default Configuration

Before configuring the card for your specific requirements Endace recommends that you return the card to the default settings using:

```
dagthree -d1 default
```

An example dagthree output is shown below:

```
linkA   noreset 10
linkB   noreset 10
packetA varlen slen=1536 align64
packetB varlen slen=1536 align64
packetA drop=0
packetB drop=0
rx      steer=stream0
tx      noifaceswap nooverlap
terf    terf_strip32
pci     33MHz 64-bit buf=128MiB rxstreams=2 txstreams=1 mem=56:16:56:0
Firmware: dag37gepci_erf_v2_5 3s1500fg456 2006/03/28 17:07:15  (user)
Card Serial: 4925 MAC Address A: 00:00:00:00:00:00 MAC Address B:
00:00:00:00:00:00 MAC Address C: 00:00:00:00:00:00 MAC Address D:
00:00:00:00:00:00
```

> **Note:** The above is an example for when `rxmerge` mode is engaged, and no transmit memory has been allocated. Although it is in `rxmerge` mode, memory has been allocated to the second memory hole. This is not strictly necessary.

Each of the items shown below can be changed

`reset` Reset the ethernet framers, set auto mode

`default` Initialise the card and set the default settings

`auto` Set autonegotiate mode, card will detect rate

`10` Force 10BaseT mode, 10Mbps

`100` Force 100BaseTX mode, 100Mbp

`1000` Force 1000BaseT mode, 1000Mbps

`slen=X` Capture X bytes of the packet content

`[no]varlen` 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 9600 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.

In fixed length mode, packets longer than the selected slen are truncated to slen, but packets shorter than slen will produce records that are padded out to the slen length.

Avoid large values of slen in fixed length mode, as short packets arriving will produce large padded records, wasting bandwidth and storage space.

`rxsplit` Send data from Port A to Stream 0.
Equivalent to Port B = Stream 2

Send data from Port B to Stream 2.
Equivalent to Port B = Stream 2

`rxmerge` Send data from Port A to Stream 0.

Send data from Port B to Stream 0.
Equivalent to Port B = Stream 0

Note: You can not change `align64`. This is permanently set.

Example

For instance, if the card is configured with fixed length capture `(novarlen)`, but configuration to variable length capture is wanted, removing or adding the "`no`" prefix will change the setting. Simply type:

```
dagthree varlen
```

```
linkA   noreset 10
linkB   noreset 10
packetA varlen slen=1536 align64
packetB varlen slen=1536 align64
packetA drop=0
packetB drop=0
rx      steer=stream0
tx      noifaceswap nooverlap
terf    terf_strip32
pci     33MHz 64-bit buf=128MiB rxstreams=2 txstreams=1 mem=56:16:56:0
Firmware: dag37gepci_erf_v2_5 3s1500fg456 2006/03/28 17:07:15  (user) Card
Serial: 4925 MAC Address A: 00:00:00:00:00:00 MAC Address B:
00:00:00:00:00:00 MAC Address C: 00:00:00:00:00:00 MAC Address D:
00:00:00:00:00:00
```

Interface Statistics

Overview

When you have configured the card according to your specific requirements you can view the interface statistics to check the status of each of the links using:

```
dagthree -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.

`Spd`	Link Speed, 10, 100 or 1000 Mbps
`Lnk`	Link state
`FD`	Full Duplex
`MA`	Device is link master
`Neg`	Auto-negotiation completed (Auto mode only)
`RF`	Remote Fault Detected Error
`JB`	Jabber Detected Error
`Err`	Ethernet Symbol Error Count

Interface Statistics (cont.)

Example

The following example is for a card with no valid input:

```
dagthree -d dag0 -si
 Spd Lnk FD Neg JB MA RF Err   Spd Lnk FD Neg JB MA RF Err
1000   0  0   0  0  1  1 65535  1000   0  0   0  0  1  1   0
1000   0  0   0  0  1  1   0  1000   0  0   0  0  1  1   0
1000   0  0   0  0  1  1   0  1000   0  0   0  0  1  1   0
```

The following is an example for a card locked to a 1000Base-T stream:

```
dagthree -d dag0 -si
 Spd Lnk FD Neg JB MA RF Err   Spd Lnk FD Neg JB MA RF Err
1000   1  1   1  0  1  0   0  1000   1  1   1  0  0  0   0
1000   1  1   1  0  1  0   0  1000   1  1   1  0  0  0   0
1000   1  1   1  0  1  0   0  1000   1  1   1  0  0  0   0
```

The following example is for a card locked to a 100base-TX stream:

```
dagthree -d dag0 -si
 Spd Lnk FD Neg JB MA RF Err   Spd Lnk FD Neg JB MA RF Err
 100   1  1   1  0  1  0   0   100   1  1   1  0  0  0   0
 100   1  1   1  0  1  0   0   100   1  1   1  0  0  0   0
 100   1  1   1  0  1  0   0   100   1  1   1  0  0  0   0
```

The following example is for a card locked to a 10base-T stream:

```
dagthree -d dag0 -si
 Spd Lnk  FD Neg  JB  MA  RF Err   Spd Lnk  FD Neg  JB  MA  RF Err
  10   1   1   1   0   1   0   0    10   1   1   1   0   0   0   0
  10   1   1   1   0   1   0   0    10   1   1   1   0   0   0   0
  10   1   1   1   0   1   0   0    10   1   1   1   0   0   0   0
```

If the RF or JB bits are 1's, this indicates a problem with the network link. This may or may not be related to the configuration of the DAG 3.7G card.

Check all cabling, ensuring that runs are not too long and that plugs are firmly clipped into their connectors. Check error condition detectors or counters on the Ethernet equipment.

Chapter 4: Capturing Data

Starting a Session

For a typical data capture session follow the steps listed below:

- Move to the `dag` directory,
- Load the appropriate driver,
- Then load the appropriate FPGA image
- Set the integrity of the card's physical layer and check the integrity of the physical layer to each DAG card. For example:

```
dagthree -d0 dag0 default
```

- Engage the failsafe relays using:

```
dagwatchdog –p –d N
```

Note: This command is not required on non-failsafe versions of the card

- Start the capture session using:

```
dagsnap -d dag0 -v -o tracefile
```

Note: You can use the `-v` option to provide user information during a capture session, although you may want to omit it for automated trace runs.

By default `dagsnap` will run indefinitely but can be stopped using *CTRL+C*. You can also configure `dagsnap` to run for a fixed time period then exit.

High Load Performance

Overview

As the DAG 3.7T 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. If not the buffer will eventually fill and packet records will be lost.

If the user process is writing records to hard disk, it may be necessary to use a faster disk or disk array. If records are being processed in real-time, a faster host CPU may be required.

In Linux and Free BSD, when the PC buffer fills, the following message displays on the PC screen:

```
kernel: dagN: pbm safety net reached 0xNNNNNNNN
```

The same message is also printed to `log /var/log/messages`. In addition, when the PC buffer fills the "Data Capture" LED on the card will flash or flicker, or may go OFF completely.

In Windows no screen message displays to indicate when the buffer is full. Please contact Endace Customer Support at support@endace.com for further information on detecting buffer overflow and packet loss in Windows.

Detecting Packet Losses

Once the buffer fills, any new packets arriving will be discarded by the DAG card until some data is read out of the buffer to create free space.

You can detect any such losses by observing the Loss Counter (`lctr` field) of the Extensible Record Format [ERF]. See *Chapter 6: Data Formats* later in this User Guide for more information on the Endace ERF.

Increasing Buffer Size

You can increase the size of the host PC buffer to enable it to cope 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. However if you are capturing at OC-12/STM-4 (622Mbps) rates or above, you may require a larger buffer.

For Linux/BSD, please refer to the *Linux FreeBSD Software Installation Guide,* which is shipped on the installation CD with the DAG3.7T card, for further information on increasing buffer size.

For Windows the upper limit is 32MB. This is usually sufficient, however if you do need to increase the amount of reserved memory please contact Endace customer support at support@endace.com for more information

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

> **Note:** For 32-bit Linux kernels, the value of `dsize` multiplied by the number of DAG cards in the system must be less than the amount of physical memory installed, as well as less than 890MB.

Packet Transmission

The firmware included with the DAG 3.7G 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 3.7G capabilities for transmitting and receiving packets.

Process	Description
Explicit packet 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 (cont.)

Process	Description
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 have a length which is a multiple of 64-bits. When capturing a packet trace for later transmission, the 64-bit alignment can be set using the `dagthree align64` command. The 64-bit alignment is permanently set on the DAG 3.7G card.

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 –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.	You can capture received traffic while transmitting. Capture programs such as `dagsnap`, `dagconvert`, and `dagbits` can be used while dagflood is sending packets.
Configuring DAG card for transmission.	To configure a DAG card for transmission, some memory must be allocated to a transmit stream. In the `dagthree` output, `buf=nMB` indicates that `n` megabytes of memory has been allocated to this DAG card in total. his 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 `dagthree 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.

In-Line Forwarding

The DAG 3.7G 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 3.7G 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.
`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. This is intended as a demonstration of Inline Forwarding technology for use in Firewall or IDS/IPS applications. It is not suitable for use as a production Firewall.
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.

Chapter 5: Synchronising Clock Time

Overview

The Endace DAG cards have sophisticated time synchronisation capabilities, which allow for high quality timestamps, optionally synchronized to an external time standard.

The core of 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. The 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 designed to reduce time variance between sets of DAG cards or between DAG cards and coordinated universal time [UTC].

You can obtain an accurate time reference by connecting an external clock to the DAG card using the time synchronisation connector. Alternatively you can use the host PCs clock in software as a reference source without any additional hardware.

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

Common Synchronization

The DAG card time 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 also provides the TDS 2 Time Distribution Server modules and the TDS 6 units that enable you to connect multiple DAG cards to a single GPS or CDMA unit.

For more information please refer to the Endace website at http://www.endace.com/accessories.htm , or the *TDS 2/TDS 6 Units Installation Manual*.

Timestamps

ERF files contains a hardware generated timestamp of each packet's arrival. The arrival time can be either the point at which the start of the packet arrives (head) or the point at which the end of the packet arrives (tail).

See *Default Configuration* in *Chapter 3: Configuring the Card* earlier in this user guide for more information on configuring the timestamp head/tail option

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

The ERF timestamp allows you to find the difference between two timestamps using a single 64-bit subtraction. You do not need to check for overflows between the two halves of the structure as you would need to do when comparing Unix time structures.

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

Below is 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;
      }
```

Configuration Tools

The DUCK is very flexible, and can be used with or without an external time reference. 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.

! **Note:** You should only run `dagclock` after you have loaded the appropriate Xilinx images. If at any stage you reload the Xilinx images you must rerun `dagclock` to restore the configuration.

```
 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                  the DAG device
  -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
```

Note: By default, all DAG cards listen for synchronisation signals on their RS-422 port, and do not output any signal to that port

```
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
```

Card with Reference

Overview

To obtain the best timestamp accuracy you should connect the DAG card to an external clock reference, such as a GPS or CDMA time receiver.

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.

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

Pulse Signal from External Source

The DAG time synchronisation connector supports an RS-422 (PPS) signal from an external source. This is derived directly from an external reference source, or distributed through the Endace TDS 2 (Time Distribution Server) module which allows two DAG cards to use a single receiver. It is also possible for more than two cards to use a single receiver by "daisy-chaining" TDS-6 expansion modules to the TDS-2 module. Each TDS-6 , module provides outputs for an additional 6 DAG cards.

Synchronise to an external source as follows:

```
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

You can connect the TDS 2 module to the DAG card using standard RJ-45 Ethernet cable including existing RJ-45 building cabling. The TDS may be located up to 600m (2000ft) from the DAG card depending upon the quality of the cable used, possible interference sources and other environmental factors. Please refer to the *TDS2/TDS6 User Guide* for more in formation

! **Caution:** Never connect a DAG card and/or the TDS 2 module to active Ethernet equipment or telephone equipment.

Testing the Signal

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

To test the signal is being received correctly and has the correct polarity use the `dagpps` tool as follows:

```
dagpps –d dag0
```

`dagpps` measures the input state many times over several seconds, displaying the polarity and length of input pulse. The DAG 3.7T card also has an LED indicator for synchronisation (PPS) signals. See *Chapter 3: Configuring the Card* earlier in this User Guide for more information.

Single Card No Reference

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 not as smooth because the PC clock is adjusted in small jumps. However the DUCK clock does not drift away from UTC.

The synchronisation achieved with this method is not as accurate as using an external reference source such as GPS.

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

```
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
```

Two Cards No Reference

Overview

If you are using two DAG cards in a single host PC with no reference clock, you must synchronise the cards using the same method if you wish to compare the timestamps between the two cards. You may wish to do this for example if the two cards monitor different directions of a single full-duplex link. You can synchronise the cards in two ways:

- One card can be a clock master for the second. This is useful if you want both cards to be accurately synchronised with each other, but not so for absolute time of packet time-stamps, or
- One card can synchronise to the host and also act as a master for the second card

Two Cards No Reference (cont.)

Synchronising with Each Other

Although the master card's clock will drift against UTC, the cards will still be locked together. This is achieved by connecting the time synchronisation connectors of both cards using a standard RJ-45 Ethernet cross-over cable.

Configure one of the cards as the master so that the other defaults to being a slave as follows:

```
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
```

Note: The slave card configuration is not shown as the default configuration will work.

Synchronising with Host

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.

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

```
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.

Connector Pin-outs

Overview

The DAG 3.7GF card has a 4-pin RJ11 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 4-pin RJ11 connector pin assignments are shown below:

1. Channel A+
2. Channel B+
3. Channel B-
4. Channel A-

Ethernet Crossover Table

You can use a standard Ethernet crossover cable to connect the two cards as shown below:

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

DAG Cards produce trace files in their own native format called ERF (Extensible Record Format). The ERF type depends upon the type of connection you are using to capture data.

The DAG 3.7T supports the following ERF Types:

ERF Type	Description
2	TYPE_ETH: Ethernet

The ERF file contains a series of ERF records with each record describing one packet.

An ERF file consists only of ERF records, there is no special file header which allows concatenation and splitting to be performed arbitrarily on ERF record boundaries.

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.

<table>
<tr><th>Byte 3</th><th>Byte 2</th><th>Byte 1</th><th>Byte 0</th></tr>
<tr><td colspan="4">timestamp</td></tr>
<tr><td colspan="4">timestamp</td></tr>
<tr><td>type</td><td>flags</td><td colspan="2">rlen</td></tr>
<tr><td colspan="2">lctr/colour</td><td colspan="2">wlen</td></tr>
<tr><td colspan="4">(rlen - 16) bytes of record</td></tr>
</table>

timestamp The time of arrival of the cell, an ERF 64-bit timestamp. See *Timestamps* in *Chapter 5: Synchronising Clock Time* earlier in this User Guide for more information on timestamps.

type One of the following:

5:	TYPE_MC_HDLC
6:	TYPE_MC_RAW
7:	TYPE_MC_ATM
9:	TYPE_MC_AAL5
12:	TYPE_MC_AAL2

Generic Header (cont.)

flags

This byte is divided into several fields as follows:

1-0: Enumerates capture interface 0-3

2: Varying record lengths

3: Truncated record (insufficient buffer space)

4: RX error (link layer error)

5: DS error (internal error)

6: Reserved

7: General direction bit. This bit has two uses, it indicates from where a packet has arrived, either the host or line, and enables the XScale to target the packet at either the host or line. The direction bit can be interpreted in the context of either the Rx or Tx hole

In the XScale/Host Rx hole, a value of "1" indicates the ERF has arrived from the line. A value of "0" indicates the record was received from the host.

In the XScale Tx hole, a value of "1" tells the ERF Mux to direct packets to the line. A value of "0" directs packets to the host.

rlen

Record length. Total length of the record transferred over the PCI bus to storage.

lctr

Depending upon the ERF type this 16 bit field is either a loss counter of colour field. The loss counter records the number of packets lost between the DAG card and the memory hole due to overloading on the PCI bus.

wlen

Wire length. Packet length including some protocol overhead. The exact interpretation of this quantity depends on physical medium.

Type 2 Record

The Type 2 Ethernet record is shown below:

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

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

Chapter 7: Troubleshooting

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.