endace
accelerated

DAG 3.6GE Card
User Guide

EDM01-04

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
support@endace.com
www.endace.com

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
support@endace.com
www.endace.com

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
support@endace.com
www.endace.com

Typographical Conventions Used in this Document

- Command-line examples suitable for entering at command prompts are displayed in `mono-space courier font`. The font is also used to describe config file data used as examples within a sentence. An example can be in more than one sentence.

  Results generated by example command-lines are also displayed in `mono-space courier font`.

- The software version references such as 2.3.x, 2.4.x, 2.5.x are specific to Endace Measurement Systems and relate to Company software products only.

Protection Against Harmful Interference

When present on product this manual pertains to and indicated by product labelling, 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.

Table of Contents

1.0 PREFACE

Introduction The installation of the Endace DAG 3.6GE 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 4.2GE Card User Manual is available on the installation CD.

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

- User Manual Purpose
- DAG 3.6GE Card Product Description
- DAG 3.6GE Card Architecture
- DAG 3.6GE Card Extended Functions
- DAG 3.6GE Card System Requirements

1.1 User Manual Purpose

Description The purpose of this DAG 3.6GE Card User Manual is to identify and describe:

- Installing DAG 3.6GE Card
- 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 3.6GE Card Product Description

The DAG Ethernet port will operate in half duplex or full duplex modes.

The DAG 3.6GE card by default finds the fastest link configuration possible with the peer device using Ethernet Autonegotiation.

Figure

Figure 1-1 shows the DAG 3.6GE series PCI card.

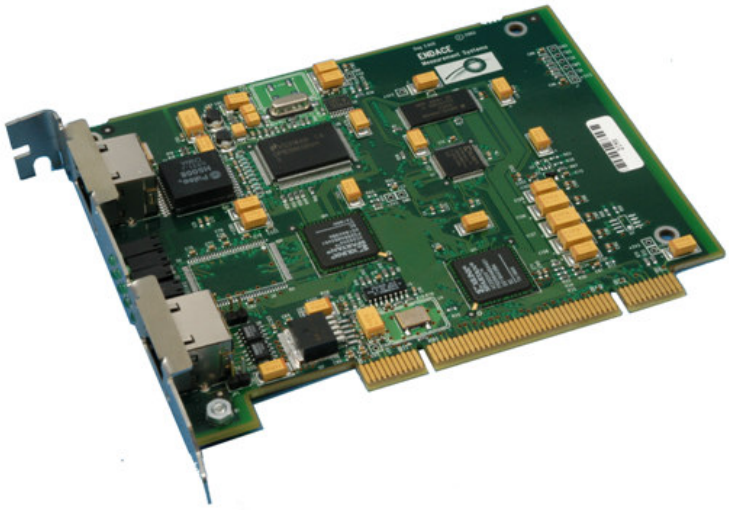

Figure 1-1. DAG 3.6GE series PCI Card.

1.3 DAG 3.6GE Card Architecture

Description

The DAG 3.6GE PCI-bus card is designed for cell and packet capture and generation on IP networks.

Serial Ethernet data is received by the interface, and fed through a framer into the first of the two Xilinx FPGAs.

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

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

Continued on next page

1.3 DAG 3.6GE Card Architecture, continued

Figure

Figure 1-2 shows the DAG 3.6GE Card major components and process flow.

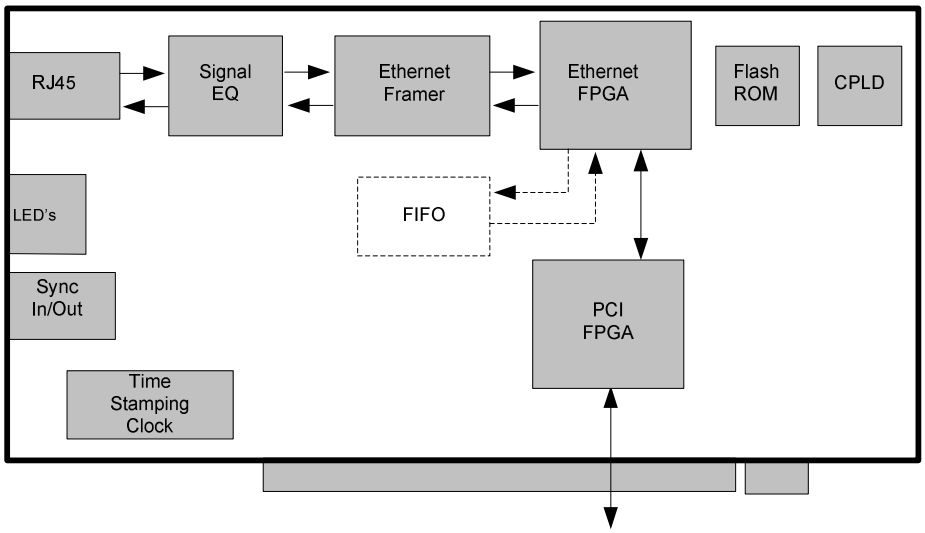

Figure 1-2. DAG 3.6GE Card Major Components and Process Flow.

DAG card as a NIC card

The DAG 3.6GE card has a single 10/100/1000 Mbps Copper Ethernet port. This is configured as if the DAG was a NIC, and can be connected to a hub, switch or router port directly.

The DAG 3.6GE can also be connected to a NIC card using an Ethernet cross-over cable. The DAG captures all packets received on this port, similar to a NIC in promiscuous mode.

1.4 DAG 3.6GE Card Extended Functions

Description

The DAG 3.6GE functionality can be extended in many ways.

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

1.5 DAG 3.6GE Card System Requirements

Description

The DAG 3.6GE and associated data capture system minimum operating requirements are:

- PC, at least Pentium II 400 MHz, Intel 440BX, GX or newer chip set
- 256 MB RAM
- At least one free PCI free slot with 3.3V and 5V power
- Software distribution free space of 30MB

Continued on next page

1.5 DAG 3.6GE Card System Requirements, continued

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

2.0 INSTALLING DAG 3.6GE CARD

Introduction The DAG 3.6GE card can be installed in any free 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.

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

- Insert DAG 3.6GE Card into PC
- Connect DAG 3.6GE Card Ports
- DAG 3.6GE Card Sensitivity

2.1 Insert DAG 3.6GE Card into PC

Description Inserting the DAG 3.6GE card into a PC involves accessing the bus slot, fitting the card, and replacing the bus slot screw.

Procedure Follow these steps to insert the DAG 3.6GE card into a PC.

Step 1. Access bus Slot

Power computer down.

Remove PCI-bus slot cover.

Step 2. Fit Card

Insert into PCI-X bus slot.

Step 3. Replace bus Slot Screw

Secure card with screw.

Step 4. Power up Computer

2.2 Connect DAG 3.6GE Card Ports

Description There are two RJ45 connectors on the DAG 3.6GE card.

The upper connector, furthest from PCI connector, is the network monitoring port. This can be connected directly to an Ethernet Hub, Switch or Router port with a standard Ethernet cable. The monitoring port can also be connected directly to a NIC card using an Ethernet cross-over cable.

The second DAG 3.6GE card RJ45 socket, near PCI connector, is for time synchronization input. This socket should never be connected to an Ethernet network or telephone line.

2.3 DAG 3.6GE Card Sensitivity

Description The DAG 3.6GE card monitoring port conforms to 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 (CAT5) cable.

By default DAG 3.6GE card automatically detects line speed of either 10, 100, or 1000Mbps.

Light link status lights indicate the network is detected correctly.

Activity lights indicate network traffic.

3.0 CONFIDENCE TESTING

Introduction The confidence testing is a process to determine the DAG 3.6GE 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 3.6GE Card LED Status
- DAG 3.6GE Card LED Display Functions
- Configuration in WYSYCC style
- DAG 3.6GE Card Capture Session
- Inspect Interface Statistics
- Reporting Problems

3.1 Interpreting DAG 3.6GE Card LED Status

Description The DAG 3.6GE has 12 status LED's with ten coloured green and two coloured orange.

On the DAG 3.6GE card the LED 3 should come on when the card is powered up.

Figure Figure 3-1 shows the DAG 3.6GE card LED locations.

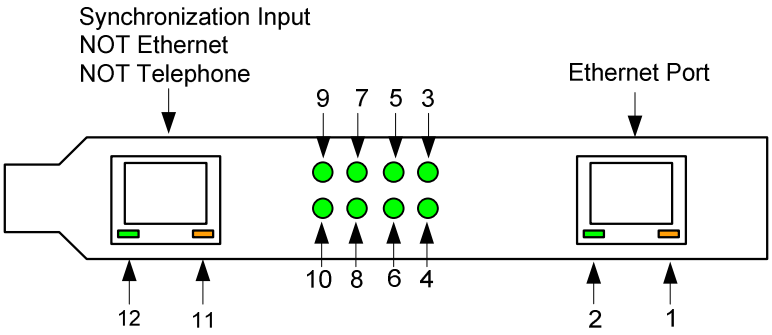

Figure 3-1. DAG 3.6GE Card LED Status LEDs.

3.2 DAG 3.6GE Card LED Display Functions

Description The function of the DAG 3.6GE card LED displays include indication of packet capture activity and links on ports A and B, and PPS signals.

Figure Figure 3-2 shows the correct LED state for DAG 3.6GE LED status on power-up with no network connection.

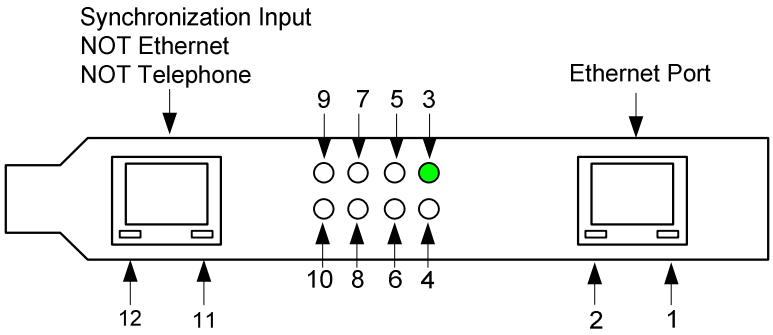

Figure 3-2. LED State for DAG 3.6GE Card With no network connection.

LED on stages The following table describes the LED display definitions:

LED	Description
LED 1	Receive activity.
LED 2	Link up.
LED 3	PCI [Lower] FPGA successfully programmed.
LED 4	PP [Upper] FPGA successfully programmed.
LED 5	Burst manager run; Indicates card is capturing packets and transferring them to the host.
LED 6	Tap mode; Always OFF on DAG 3.6 GE.
LED 7	10Base-T Link Up
LED 8	100Base-TX Link Up
LED 9	1000Base-T Link Up
LED 10	Reserved.
LED 11	PPS Out: Pulse Per Second Out; Indicates card is sending a clock synchronization signal.
LED 12	PPS In: Pulse Per Second Out; Indicates card is receiving an external clock synchronization signal.

Continued on next page

3.2 DAG 3.6GE Card LED Display Functions, continued

Configuration utility

The `dagthree` utility supports configuration status and physical layer interface statistics for the DAG 3.x series of cards.

In a troubleshooting configuration options `-si` should be passed to the tool to watch physical and framing layers operational status.

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

3.3 Configuration in WYSYCC style

Description

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

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

Configuration options

`reset`	reset the ethernet framers, set auto mode
`default`	reset the ethernet framer, set auto mode
`auto`	set autonegotiate mode, card will detect rate
`10`	force 10BaseT mode, 10Mbps
`100`	force 100BaseTX mode, 100Mbps
`1000`	force 1000BaseT mode, 1000Mbps
`[no]varlen`	dis/enable variable length capture. Otherwise record length padded to slen.
`slen=X`	capture X bytes of the packet content

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

```
dag@endace:~$ dagthree -d dag0 varlen
link    noreset auto enablea
packet  varlen slen=48
packetA drop=0
pci     33MHz 32-bit buf=32MB rxstreams=1 txstreams=0
mem=32:0
```

Once the card has been configured the interface statistics are inspected to check the card has correctly detected the links.

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

3.4 DAG 3.6GE Card Capture Session

Description A successful DAG 3.6GE card capture session is accomplished by checking the card has correctly detected the link. This is followed by configuring the DAG card for normal use.

Procedure Follow these steps for a successful DAG 3.6GE card capture session.

Step 1. Check Cabling

Ensure cabling is correctly connected and that RJ45 connectors are clipped into the sockets.

Step 2. Understand link layer configuration

Learn about the link layer configuration in use at the network link being monitored.

Important parameters include specific scrambling options in use.

If the information cannot be obtained reliably, the card can be made to work by varying the parameters until data is arriving at the host system.

Step 3. Check FPGA Images are Loaded

Ensure the most recent pair of FPGA images have been loaded onto the card. The link status and activity LEDs will not activate until the upper FPGA firmware is downloaded.

```
dag@endace:~$ dagrom -rvp -d dag0 -f xilinx/dag36epci-erf.bit

dag@endace:~$ dagld -x -d dag0 -f xilinx/dag36gepp-erf.bit

dag@endace:~$ dagthree -d dag0
link    noreset auto enablea
packet  novarlen slen=48 noalign64
packetA drop=0
pci  33MHz 32-bit buf=32MB rxstreams=1 txstreams=0 mem=0:0
```

NOTE: The `dagld` step has been missed if the card read-out looks like this:

```
dag@endace:~$ dagthree -d dag0
link    noreset 100
packet  novarlen slen=1128481603 noalign64
packetA drop=1229539657
pci    33MHz 32-bit buf=32MB rxstreams=1 txstreams=0 mem=0:0
```

Continued on next page

3.4 DAG 3.6GE Card Capture Session, continued

Procedure,continued

Step 4. Configure DAG 3.6GE Card for normal use

The `dagthree default` command is always used:

```
dag@endace:~$ dagthree default
link    noreset auto enablea
packet  novarlen slen=48 noalign64
packetA drop=0
pci   33MHz 32-bit buf=32MB rxstreams=1 txstreams=0 mem=32:0
```

3.5 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:~$ 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

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

```
dag@endace:~$ dagthree -d dag0 -si
 Spd   Lnk  FD  MA  Neg  RF  JB
1000   0   0   0   0   0   0
1000   0   0   0   0   0   0
1000   0   0   0   0   0   0
```

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

```
dag@endace:~$ dagthree -d dag0 -si
 Spd   Lnk  FD  MA  Neg  RF  JB
 100   1   1   0   1   0   0
 100   1   1   0   1   0   0
 100   1   1   0   1   0   0
```

Continued on next page

3.5 Inspect Interface Statistics, continued

Description,continued

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

```
dag@endace:~$ dagthree -d dag0 -si
 Spd   Lnk  FD  MA   Neg   RF   JB
 100    1    1   0    1     0    0
 100    1    1   0    1     0    0
 100    1    1   0    1     0    0
```

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

```
dag@endace:~$ dagthree -d dag0 -si
 Spd   Lnk  FD  MA   Neg   RF   JB
1000   1    1   0    1     0    0
1000   1    1   0    1     0    0
1000   1    1   0    1     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 3.6GE 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.

3.6 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: `dagthree`
10.	Network link statistics reported by: `dagthree -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 captured packet trace illustrating the problem.

4.0 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

4.1 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
Setting capture session parameters	Parameters are set with `dagthree`. 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 2044 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/dagthree -d dag0 varlen slen=1536`

Continued on next page

4.1 Starting Capture Session, continued

Process,continued

Process	Description
Setting fixed length mode.	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. Large values of slen in fixed length mode, as short packets arriving will produce large padded records, wasting bandwidth and storage space. For fixed length 64-byte records for example, choose `slen=44` (64 – ERF header size of 16 – alignment padding 4): `dagthree –d dag0 novarlen slen=44`
Setting packet capture settings.	Capture settings must be set for each card in use. A capture session is started on a card, using dagsnap. `dagsnap –v –o tracefile` The option `-v` is used to provide user information during capture; it can be omitted for automated trace runs. If the `tracefile` parameter is not specified the tool will write to stdout, which can be used to pipeline `dagsnap` with other tools from the `dagtools` package.
Stopping dagsnap running.	By default `dagsnap` will run 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 with the `-s option`.

4.2 High Load Performance

Description

As the DAG 3.6GE 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 3.6GE 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 3.6GE card Windows operating system the upper limit is 32MB.

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.

5.0 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.

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

5.1 Configuration 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 (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 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
```

5.2 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

5.2.1 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 synchronise 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.

5.2.2 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 synchronise 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.

Continued on next page

5.2.2 Two Cards no Reference Time Synchronization, continued

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.

5.2.3 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.

Continued on next page

5.2.3 Card with Reference Time Synchronization, continued

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 synchronises 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.

5.3 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 Figure 6-1 shows the RJ45 plug and socket connector pin-outs.

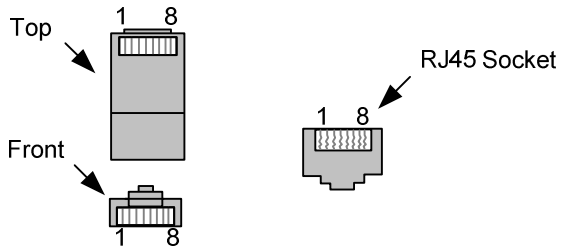

Figure 6-1. RJ45 Plug and Socket Connector Pin-outs.

Continued on next page

5.3 Synchronization Connector Pin-outs, continued

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 cards 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.

6.0 DATA FORMATS OVERVIEW

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

- Data Formats
- Timestamps

6.1 Data Formats

Description The DAG 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.

Table Table 7-1 shows the generic variable length record.

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

Continued on next page

6.1 Data Formats, continued

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.

Continued on next page

6.1 Data Formats, continued

Table Table 7-2 shows the Type 2 Ethernet variable length record. The diagram is not to scale.

<table>
<tr><td colspan="3">timestamp</td></tr>
<tr><td colspan="3">timestamp</td></tr>
<tr><td>type:2</td><td>flags</td><td>rlen</td></tr>
<tr><td colspan="2">lctr</td><td>wlen</td></tr>
<tr><td>offset</td><td>pad</td><td>rlen-18</td></tr>
<tr><td colspan="3">bytes of frame</td></tr>
</table>

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.

Continued on next page

7.2 Timestamps, continued

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;
      }
```