



DAG 3.7D
Card User Guide
EDM01-16

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

Copyright 2006 ©All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher.

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.

Table of Contents

Chapter 1: Introduction	1
Overview	1
Purpose	1
Card Description	2
Card Architecture	2
Extended Functions	3
System Requirements	4
Chapter 2: Installation	5
Introduction	5
DAG Device Driver	5
Inserting the Card	5
Port Connectors	5
Chapter 3: Confidence Testing	8
Introduction	8
Card Sensitivity	8
LED Status	9
LED Display Functions	9
Dagthree Utility	9
Card ConfigurationOptions	10
Interface Statistics	11
Status Bits Display	11
Card Capture Session	12
Reporting Problems	14
Chapter 4: Running Data Capture Software	15
Starting a Capture Session	15
High Load Performance	15
Chapter 5: Synchronizing Clock Time	17
Introduction	17
Configuration Tools	18
Single Card No Reference	19
Two Cards No Reference	20
Card with Reference	21
Connector Pin-outs	23
Chapter 6: Data Formats	25
Overview	25
Generic Header	25
Pos HDLC Record	26
ATM Cell record	26
Timestamps	27

Chapter 1: Introduction

Overview

The installation of the Endace DAG 3.7D 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.

The characteristics include card architecture, extended functions, and system requirements.

This document, DAG Card 3.7D User Guide is available when the installation CD is placed in a running Windows PC.

Purpose

Description

The purpose of this DAG Card installation guide is to describe:

- Installing the DAG3.7D
- Confidence Testing
- Running Data Capture
- Synchronising 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 3.0 r4 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 EDM04-01 Endace Software Installation Manual.

To install on a Windows operating system, follow the instructions in the document EDM04-02 Endace Windows Installation Manual.

Card Description

The DAG cards are PCI-bus cards designed for cell and packet capture and generation on IP networks.

The DAG 3.7D card has dual DS3/E3 co-axial interface, both receive and transmit and is shown below:



Card Architecture

Description

Because of component close association, packets or cells are time-stamped accurately. Time stamped packet records are stored in the FPGA, which interfaces to the PCI bus.

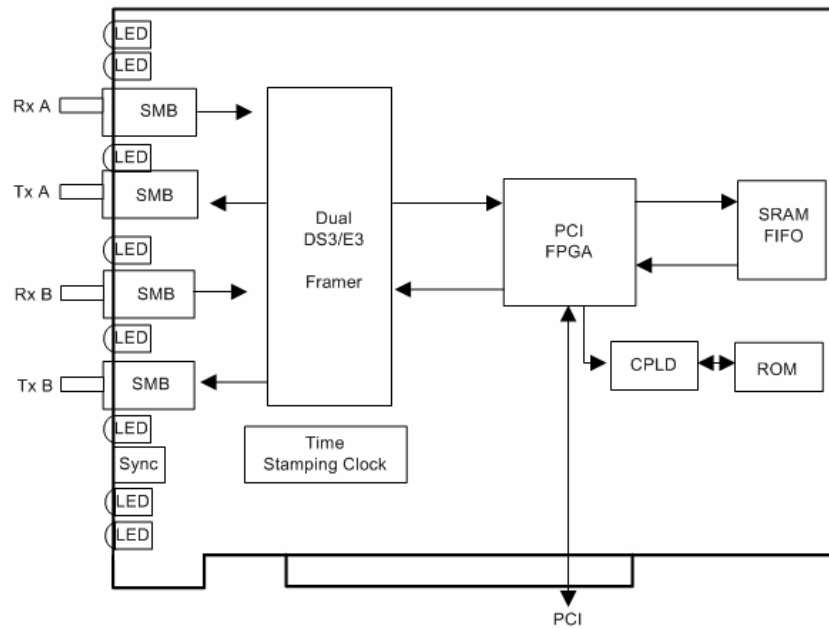
All packet records are written to host PC memory during capture operations.

Serial PDH data is received by the DAG 3.7D card co-axial interfaces, and fed into dual DS3/E3 framer.

Network data is then fed to the FPGA which also contains the DUCK timestamp engine. The close association of these two components means that packets or cells can be time-stamped very accurately.

Card Architecture (cont.)

The diagram below shows the DAG 3.7D card major components and process flow.



Time stamped packet or cell records are stored in a FIFO.

The DAG 3.7D can demap either ADM (ATM Direct Mapped) ATM cells, PLCP (Payload Layer Convergence Protocol) ATM cells, PPP encapsulations, HDLC (High-Level Data Link Control) and Frame Relay.

The DAG3.7D also supports subrate DS3 for Kentrox data encapsulation formats but requires a specific image to do so. Please contact Endace Customer Support at support@endace.com for more information.

Records are then transferred from the FIFO into the FPGA, which has interfaces to the PCI bus.

The current firmware only supports bit synchronisations HDLC demapping.

Extended Functions

The functionality of the DAG 3.7D card can be extended in many ways. A physical transmit path is provided on the card so that packet organisation is possible, requiring special firmware images.

The framer is normally set up to map DS3 payloads, but other mappings are possible.

To discuss the use of other extended features contact the Endace customer support team at support@endace.com

System Requirements

General

The DAG 3.7D card and associated data capture system minimum operating requirements are:

- PC, at least Pentium II 400 MHz, Intel 440BX, GX or newer chip set
- Minimum of 128 MB RAM
- At least two free PCI free slot with 3.3V and 5V power
- Software distribution free space of 30MB
- Endace Linux Install CD requires 6GB

Operating System

For convenience, a 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, RHEL 4.0 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

A DAG 3.7D card can be installed in any free 3.3V or 5V Bus Mastering PCI slot. The card should be the only device on the PCI bus if possible as the cards make very heavy use of PCI 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. However, 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

If the DAG device driver is not installed, before proceeding with the next chapter, install the software by following the instructions in EDM04-01 Endace Software Installation Manual.

Inserting the Card

Inserting the DAG 3.7D card into a PC involves accessing the PCI-X bus slot, fitting the card, and replacing bus slot cover. 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 four metal co-axial SMB connectors on the DAG 3.7D card.

The bottom connector of each pair is used for the transmitted signal, the top for the received signal.

Note: The transmit ports only require connection if the transmit feature is supported by the firmware, or if the loopback facility on the DAG is enabled. For passive monitoring you do not need to connect the transmit ports.

Packet transmission is not currently supported.

The DAG 3.7D card has an 8-pin RJ45 socket for the time synchronization input.



WARNING: Do not connect the socket to an Ethernet.

Chapter 3: Confidence Testing

Introduction

Confidence testing is a process to determine the DAG 3.7D card is functioning correctly.

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

Interface statistics are also inspected during this process.

Card Sensitivity

Overview

The input signal level to the DAG 3.7D card should be within the dynamic range of the receiver. If the input signal power is slightly out of range then an increased bit error rate will be experienced.

If the power is well out of range then the system will be unable to lock to the PDH frames.

Signal Source Specifications

The signal source should meet DS3 template of ANSI-T102.1993 Figure 4 and STS-1 template of ANSIT102.1993 Figure 5, Loss characteristics of the WE728A or RG 59B cable should be better than Figure C2 of ANSI-T102.1993.

Receiver Capability

The receiver can handle up to 450 feet of cable loss (5.5dB) from the DSX cross-connect.

Input Signal Level

The input signal level is measured in mVp, the Peak Differential Input Amplitude in millivolts.

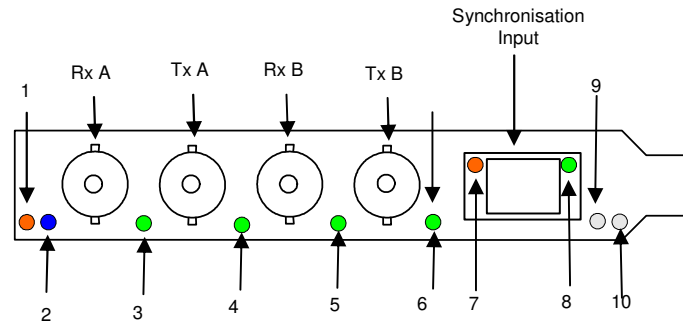
There is an optional RX-monitor mode that adds 20dB gain. This is intended for use with DS3 Monitor signal ports. The DS3 Monitor port is a resistively attenuated (around -21.5dB) copy of the DSX3 cross connect signal. The minimum signal level at a DSX3 cross-connect is 360mVp, so the DSX3 Monitor level at minimum is 30mVp, requiring the RX-monitor mode.

Optimum Setup

The optimum way to set up a DAG card is to measure the signal level at the receiver, and to make sure that it is well within the specified range.

LED Status

The DAG 3.7D card has eight status LED's, five coloured green, two orange and one blue as shown below:



The LED definitions are shown below:

LED	Description
1	Burst Manager Run (orange)
2	FPGA successfully programmed . (blue)
3	Signal Detect Port A (green.
4	Not enabled
5	Signal Detect Port B (green)
6	Not enabled
7	PPS out,.
8	PPS in, . indicates card is receiving a PPS signal (green)
9	Not enabled
10	Not enabled

LED Display Functions

On the DAG 3.7D card LED 8 flashes when a PPS signal is being received, LED 7 flashes when a PPS signal is being outputted.

When DS3 signals are applied LEDs 3 and 5 should come on. The status of these LEDs should not change during normal operation of the card.

LED 1 lights when a capture is in progress.

The correct LED state for the DAG 3.7D card without input signal is shown below:

Dagthree Utility

The `dagthree` utility supports configuration status and physical layer interface statistics for the DAG 3.7D card.

When troubleshooting, options `-si` should be passed to the tool to watch the operational status of the physical, PDH and framing layers. More details about the meaning of the various bits are supplied through the help page (`dagthree -h`) as well as via the manual page.

Card Configuration Options

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

default	set card to normal defaults
[no]fcl	(un) set facility Loopback. Should be set to nofcl. Mainly used for testing
[no]eql	(un)set equipment loopback. Should be set to noeql. Mainly used for testing
[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
rxonly	Assign all buffer memory to receive streams
[no]rxmonitor	Used with DS3X monitor tap. Enabling the pre-amplifier adds approx 20dB of linear amplification
[no]descramble	(dis)enable receive cell scrambling. ATM only
(no)discard	When enabled the FCS bytes are discarded. When disabled the FCS bytes are passed on. This option is ignored when nocrc is set
nocrc/crc16/crc32	Sets the expected size of the crc for the receive packet to process. HDLC only
ds3_cbit	Framing DS3 CBit
ds3_cbit_if	Framing DS3 CBit Internal Fractional
ds3_cbit_ef	Framing DS3 CBit External Fractional
ds3_cbit_ff	Framing DS3 CBit Flexible Fractional
ds3_m23	Framing DS3 M23
ds3_m23_if	Framing DS3 M23 Internal Fractional
ds3_m23_ef	Framing DS3 M23 External Fractional
ds3_m23_ff	Framing DS3 M23 Flexible Fractional
Hdlc	Selects bit-synchronous HDLC mapping. This is currently the only option supported

Interface Statistics

Once the DAG 3.7D card has been configured as expected, the interface statistics should be inspected to see if the card is locked to the data stream.

```
port A      dag@endace:~$ dagthree -d0 -asi
port B      dag@endace:~$ dagthree -d0 -bsi
both ports  dag@endace:~$ dagthree -d0 -si
```

Status Bits Display

The tool will display a number of status bits as they have occurred since the last time read. In our example, the interval is set to one second via the `-i` option.

LoS	DS3 Loss of Signal. This indicates that there is no signal at the receiver or signal strength is too low to be recognized.
AIS	Framer is declaring AIS.
OoF	DS3 Out of Framing. The DS3 Framer is not synchronised.
Idle	Framer is detecting idle pattern in its incoming stream.
RLOL	Receive loss of lock
LOF	Loss of frame
RDI	Remote defect indication.

An example of a card locked to a DS3 ATM stream is shown below::

```
A:   LoS  AIS  OoF  Idle
      0    0    0    0
      0    0    0    0
      0    0    0    0
```

An example of when there is no valid signal input is shown below:

```
A:   LoS  AIS  OoF  Idle
      0    0    0    0
      0    0    0    0
      0    0    0    0
```

Card Capture Session

Overview

The DAG 3.7D card uses the OS3/E3 ATM UNI/PPP physical layer interface device to support capturing of ATM. The card supports the DS3 standard.

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

Follow the steps shown below to troubleshoot DAG card configuration.

Check Receiver Port Signal Levels

- Ensure that the signal levels reaching the receiver ports of the card are at the right levels.

Understand Link Layer Configuration

- Learn about the link layer configuration in use at the network link being monitored. Important parameters include m23 Vs. c-bit framing, as well as the mapping 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.

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.

List Current Settings

- For DAG 3.7D framer configuration and statistics the `dagthree` tool is supplied.
- Calling `dagthree` without arguments lists current settings.
- The `dagthree -h` prints a help listing on tool usage.

Check FPGA Image Loaded

- Before configuring the card, ensure the most recent FPGA image is loaded on the card by loading the newest image using
`dagrom -RVP -f <xilinx/dag37pci_erf.bit:`

Card Capture Session (cont.) **Add -f flag**

- Add the `-f` flag to `dagld`.
`dag@endace:~$ dagthree -d0`

```
PortA: rx_monitor ds3_m13 nofcl hdlc
PortB: rx_monitor ds3_m13 nofcl hdlc
packet varlen slen=64 align64
packetA drop=0
packetB drop=0
pci      33MHz 32-bit nodrop routesource=stream0
buf=128MiB rxstreams=1 txstreams=0 mem=128:
```

Configure in WYSYCC Style

If the card is configured for c-bit (`ds3_cbit`) framing mode but m13/23 framing is required, changing the suffix will alter the framing mode type:

```
dag@endace:~$ dagthree -d /dev/dag0 ds3_m13

PortA: rx_monitor ds3_m13 nofcl hdlc
PortB: rx_monitor ds3_m13 nofcl hdlc
packet varlen slen=64 align64
packetA drop=0
packetB drop=0
pci      33MHz 32-bit nodrop routesource=stream0
buf=128MiB rxstreams=1 txstreams=0 mem=128:
```

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

Starting a Capture Session

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

For a typical measurement session follow the following steps:

- First move to the `dag` directory,
- Load the driver,
- Then load the appropriate Xilinx receive image to each DAG card. For example, with the DAG 3.7D card installed:

```
dagrom -ryvp -f xilinx/dag37dpci_erf.bit
```

The integrity of the card's physical layer is then set and the integrity of the physical layer to both DAG cards checked.

Follow the process below to set a data capture session:

- Set and check integrity of card physical layer. For example:

```
dagthree -d0 dag0 default
```
- Start capture session using:

```
tools/dagsnap d0 -v -o tracefile
```

Note: The option `-v` is used to provide user information during capture; it may be wanted to omit it for automated trace runs. If the `-o 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.

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 then exit.

High Load Performance

Overview

As the DAG 3.7D 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.

When the PC buffer fills, the message `kernel: dagN: pbm safety net reached 0xNNNNNNNN` 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.

High Load Performance (cont.)

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 is detected in-band by observing the Loss Counter `lctr` field of the Extensible Record Format [ERF]. See Chapter 6 of this Guide for more information on the Endace ERF.

Reading Records

To avoid any potential packet loss, the user process must read records faster than they arrive from the network.

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.

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 3.7T card Windows operating system the upper limit is 32MB.

In Debian Linux the amount of memory reserved is changed by editing 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.

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 PC's 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.

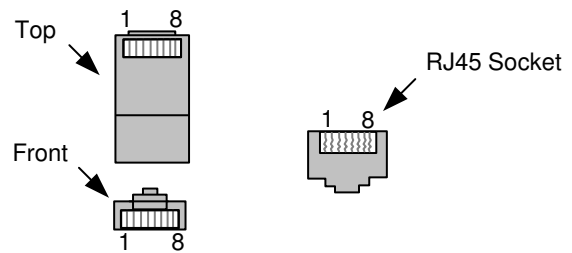
Connector Pin-outs Overview

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

The DAG card produces trace files in the Endace Extensible Record Format (ERF). The ERF consists of a series records. Each record describes one packet.

An ERF file consists only of ERF records. There is no special file header. This 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.

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	1 : TYPE_HDLC_POS (PoS w/HDLC framing), or 3 : TYPE_ATM (ATM cell) .
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.

Pos HDLC Record

The Type 1 PoS HDLC Record format is for HDLC data links. For example:

- Packet over SONET
- Point-to-Point Protocol [PPP] over SONET
- Frame Relay

The record is described below:

BYTE 3	BYTE 2	BYTE 1	BYTE 0
timestamp			
timestamp			
type 1	flags	rlen	
lctr		wlen	
HDLC Header			
(rlen - 20) bytes of record			

ATM Cell record

The record format for the ATM cell capture is the Type 3 ATM cell record. ATM Header, it does not include the 8-bit HEC.

The record is shown below:

BYTE 3	BYTE 2	BYTE 1	BYTE 0
timestamp			
timestamp			
type 3	flags	rlen	
lctr		wlen	
ATM Header			
48 bytes of cell			

Timestamps

The ERF 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 user 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 codes

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;
}

```