endace
accelerated

DAG 3.5S Card
User Guide

EDM01-13v14

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 2001
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 OAX
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`.

  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 3.5S 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 3.5S Card Product Description
- DAG 3.5S Card Architecture
- DAG 3.5S Card System Requirements

1.1 User Manual Purpose

Description The purpose of this DAG 3.5S Card User Manual is to identify and describe:

- Installing DAG 3.5S card
- Setting DAG 3.5S Card Optical Power
- Confidence Testing DAG 3.5S Card
- 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 3.5S Card Product Description

Description The DAG cards are PCI-bus cards designed for cell and packet capture and generation, specialised on IP networks. Various versions have been produced with different interfaces, this manual documents the:

DAG 3.51 OC3/12 optical interface
DAG 3.52: OC3/12 optical interface

Figure Figure 1-1 shows the DAG 3.5S card.

Figure 1-1. DAG 3.5S Card.

1.3 DAG 3.5S Card Architecture

Description Serial SONET optical data is received by the optical interface, and fed through a demultiplexor into the upper of two Xilinx FPGAs.

This FPGA contains a SONET framer and the DUCK timestamp engine.

The close association of these two components means that packets or cells can be time-stamped very accurately. Time stamped packet or cell records are then stored in a FIFO.

Continued on next page

1.3 DAG 3.5S Card Architecture, continued

Figure Figure 1-2 shows the DAG 3.5S card major components and data flows.

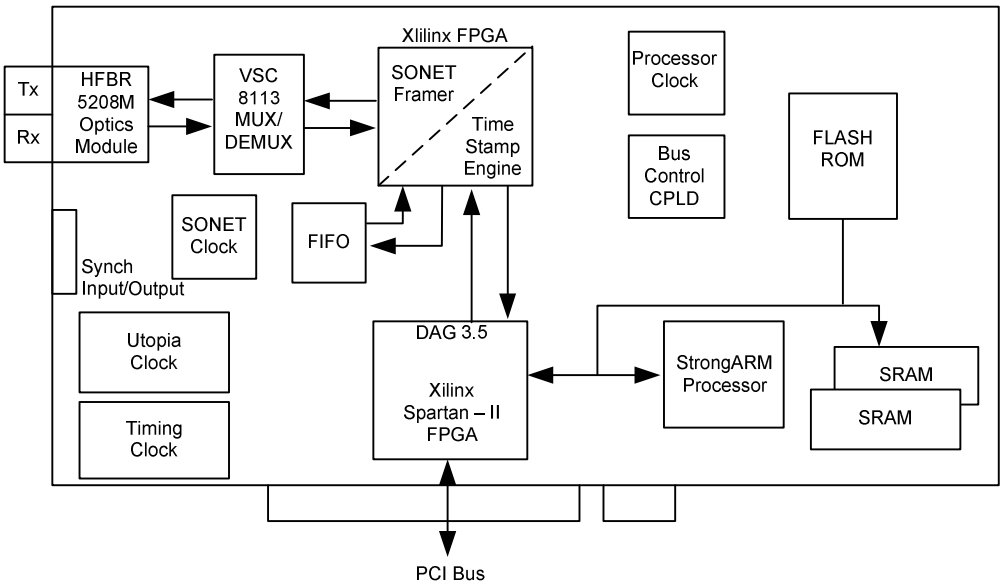

Figure 1-2. DAG 3.5S Card Major Components and Data Flows.

Records are transferred from the FIFO into the lower FPGA, which has interfaces to the PCI bus and to the bus of the local StrongARM processor.

The entire record is made available to the StrongARM processor in a series of memory-mapped registers. Code running on the StrongARM is then used to decide if the record should be written to host PC memory over the PCI bus.

In the case of ATM the StrongARM code can be configured to copy the first n, or all cells, of each AAL5 SDU. For POS normally all packet headers are copied. New images available for POS allow the capture of a variable amount of the packet.

The functionality of the DAG card 3.5S can be extended in many ways. The user can develop StrongARM code to filter records, or to collect statistics. The framer is normally set up to map STM-1, STS-3c, STM-4c and STS-12c payloads, but other mappings are possible.

A transmit path is provided on the DAG 3.5S card, so cell or packet generation is possible, but this requires special FPGA images. For detailed information on device address mapping on the StrongARM bus, or to discuss the use of other extended features please contact: support@endace.com.

1.4 DAG 3.5S Card System Requirements

Description

The DAG 3.5S card and its associated data capture system has the following minimum system requirements:

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

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, 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 3.5S CARD

Introduction A DAG 3.5S card can be installed in any free Bus Mastering PCI slot.

By default, the driver supports up to four DAG cards in one system, but it is not recommended to have more than 2 cards on a single PCI bus due to bandwidth limitations, as 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.

- Installation of Operating System and Endace Software
- Insert DAG 3.5S Card into PC
- DAG 3.5S Card Port Connectors
- Pluggable Optical Transceivers

2.1 Installation of Operating System and Endace Software

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

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

2.2 Insert DAG 3.5S Card into PC

Description Inserting the DAG 3.5S card into a PC involves accessing the PCI bus slot, fitting the card, and replacing bus slot screw.

Procedure Follow these steps to insert the DAG 3.5S card into a PC.

Step 1. Access bus Slot

Power computer down.

Remove PCI bus slot cover.

Step 2. Fit Card

Insert DAG 3.5S card into PCI bus slot.

Ensure free end fits securely into a card-end bracket that supports the card weight.

Continued on next page

2.2 Insert DAG 3.5S Card into PC, continued

Procedure, continued

Step 3. **Replace bus Slot Screw**

Secure card with screw.

Step 4. **Power Up Computer**

2.3 DAG 3.5S Card Port Connectors

Description

There are two square SC-type optical connectors on the DAG 3.5S card. The bottom connector, nearest the PCI slot, is for the received signal. The top connector is for the transmitted signal.

Connection need only be to the transmit port if using the loop back facility in the DAG card to daisy chain systems, or if a data generation program is being used.

If the DAG card Tx port of the is not used, the SC-type transceiver optics should be covered to prevent ingress of dust.

The DAG 3.5S card has an 8-pin RJ45 socket for the time synchronization input, it is never connected to an Ethernet. For details of pin outs see Chapter 11.2.

8-pin RJ45 socket

The DAG 3.5S card has an 8-pin RJ45 socket for the time synchronization input.

CAUTION: Do not connect the socket to an Ethernet.

3.0 SETTING DAG 3.5S CARD OPTICAL POWER

Introduction

The optical power range depends on the particular transceiver module fitted to the DAG 3.5S card.

The power range depends on the particular device installed on a DAG card. The DAG 3.5S DAG card is shipped fitted with HFBR 5208M module by default.

Optical power measure

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

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

Configuration

The following table describes the DAG 3.5S card optics power module configuration. MMF = Multi Mode Fibre. SMF = Single Mode Fibre.

Part No.	Fibre	Data Rate	Max Power [dBm]	Min Power [dBm]	Nominal Pwr [dBm]
HFBR 5208M	MMF	155/622	-14	-26	-20
HFBR 5208	MMF	155/622	-14	-26	-20
HFCT 5208M	SMF	155/622	-7	-28	-20

In this chapter

This chapter covers the following sections of information.

- DAG 3.5S DAG Card Optical Power Output
- Splitter Losses

3.1 DAG 3.5S DAG Card Optical Power Output

Description

The optical power input to the DAG 3.5S card should be within the dynamic range of the receiver.

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

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

Continued on next page

3.1 DAG 3.5S DAG Card Optical Power Output, continued

Input power

The DAG 3.5S card is set up to measure the optical power at the receiver, and to make sure that it is well within the specified power range.

Input power is adjusted by:

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

3.2 Splitter Losses

Description

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

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

Single mode fibre loss

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

Multi-mode fibre loss

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

4.0 CONFIDENCE TESTING DAG 3.5S CARD

Introduction

The confidence testing is a process to determine the DAG 3.5S 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 3.5S Card LED Status
- DAG 3.5S Card LED Display Functions
- DAG 3.5S Card Capture Session
- Configuration in WYSYCC Style
- DAG 3.5S Card Configuration Options
- Verify DAG 3.5S Card Configuration
- Reporting Problems

4.1 Interpreting DAG 3.5S Card LED Status

Description

The DAG 3.5S card has 8 status LEDs, five green and three red.
There are two sets of Xilinx images for the DAG 3.5S card, with different LED assignments.

The Legacy format images perform fixed length capture. The ERF format images add variable length capture features.

The DAG 3.51 has the Legacy Format images in ROM, the DAG 3.52 has the ERF Format images in ROM. The ERF Format images may be downloaded to any DAG 3.5S card to enable variable length capture.

Figure

Figure 4-1 shows the DAG 3.5S card status LEDs.

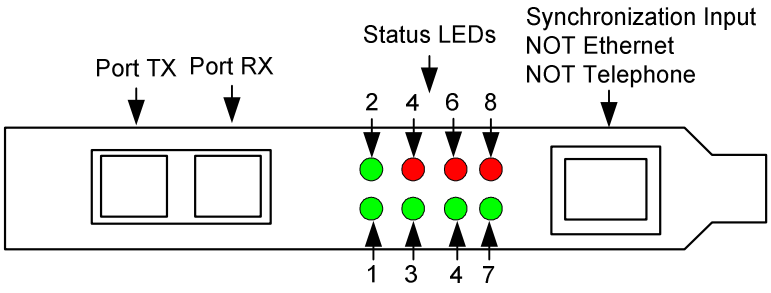

Figure 4-1. DAG 3.5S Card Status LEDs.

Continued on next page

4.1 Interpreting DAG 3.5S Card LED Status, continued

LED definitions The following table describes the Legacy Format images used to describe the state when the LED is ON.

LED	Display Function
LED 1	Upper FPGA successfully programmed.
LED 2	Lower FPGA successfully programmed.
LED 3	Flashes once per second to indicate proper device operation.
LED 4	User LED, driven by a register in the StrongARM address space.
LED 5	LOS: Loss of signal – no signal seen by the optical receiver.
LED 6	LOF: Loss of frame synchronization – usually caused either by loss of signal, or by the system being set to the wrong OC rate.
LED 7	LOP: loss of pointer lock – usually caused by the system being set to the wrong OC rate.
LED 8	LCD: loss of cell delineation, will always be on for POS, but should go out when the device synchronizes with the cell stream for ATM.

ERF format images The following table describes the ERF Format images.

LED	Display Function
LED 1	Upper FPGA successfully programmed.
LED 2	Lower FPGA successfully programmed.
LED 3	Burst Manager Run – Indicates the card is capturing packets and transferring them to the host
LED 4	ATM: This LED is on if card is configured for ATM capture.
LED 5	PPS: Pulse Per Second – indicates the card is receiving an external clock synchronization signal.
LED 6	LOF: Loss of frame synchronization – usually caused either by loss of signal, or by the system being set to the wrong OC rate.
LED 7	LOP: loss of pointer lock – usually caused by the system being set to the wrong OC rate.
LED 8	LCD: loss of cell delineation, will always be on for POS, but should go out when the device synchronizes with the cell stream for ATM.

4.2 DAG 3.5S Card LED Display Functions

Description The LED display functions are relative to the legacy format images and the ERF format images.

When the DAG 3.5S series of cards is powered up the 1 and 2 LEDs should come on.

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

- DAG 3.5S Card LED Display for Legacy Format Images
- DAG 3.5S Card LED Display for ERF Format Images

4.2.1 DAG 3.5S Card LED Display for Legacy Format Images

Description For Legacy Format images LED 3 starts flashing.

When an optical signal is applied LEDs 5 and 6 should go out.

LED 8 should stay on for POS, but go out for ATM.

The status of these LEDs should not change during normal operation of the card.

Figure Figure 4-3 shows the correct LED state for the DAG 3.5S card with fixed length capture images without optical input.

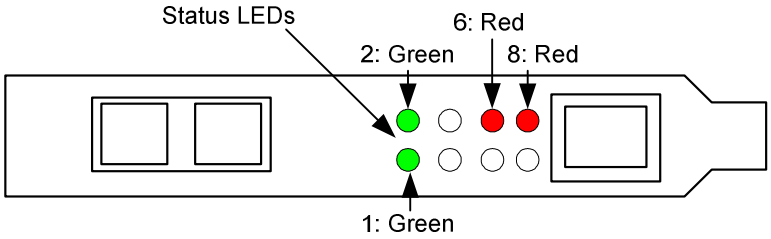

Figure 4-3. Correct LED State for DAG 3.5S Card With Fixed Length Capture Images Without Optical Input.

4.2.2 DAG 3.5S Card LED Display for ERF Format Images

Description

For ERF Format images LEDs 3 and 4 are off.

No LED flashes unless an external clock synchronization cable is connected to the RJ45 socket. LED 6, 7 and 8 are as before.

Figure

Figure 4-4 shows the correct LED state for the DAG 3.5S card with variable length capture images without optical input.

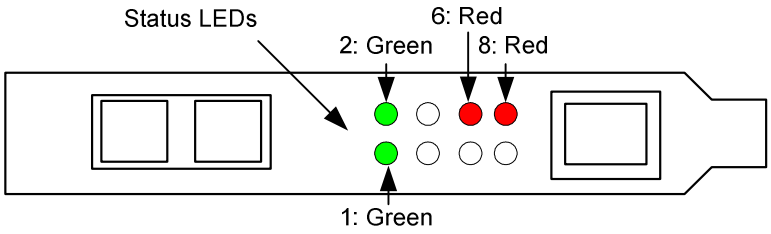

Figure 4-4. Correct LED State for DAG 3.5S Card With Variable Length Capture Images Without Optical Input.

`dagthree` utility

The `dagthree` utility supports configuration status and physical layer interface statistics for the DAG 3.5S series of cards.

In a troubleshooting configuration options `-si` should be passed to the tool to watch the operational status of the optical, SONET 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.

4.3 DAG 3.5S Card Capture Session

Description

The DAG 3.5S card uses an ASIC SONET ATM/PoS physical layer interface device to support capturing of ATM cells and HDLC encoded Packet-over-SONET data frames. The card supports both OC3c and OC12c standards.

Because of its flexibility, the correct link layer configuration needs to be supplied to the card to function as expected.

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 a DAG card for normal use.

Continued on next page

4.3 DAG 3.5S Card Capture Session, continued

Procedure Follow these steps to configure the DAG 3.5S card.

Step 1. Check Receiver Ports Optical Signal Levels.

The card supports 1300 nanometer single-mode fibre attachments with optical signal strength between -11 dBm and -29 dBm.

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

The card receiver port is the lower of the dual-SC-style connector, the closest to the LED's.

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

Step 2. Understand Link Layer Configuration

Learn about the link layer configuration in use at the network link being monitored. Important parameters include OC3c vs. OC12c configuration, ATM vs. PoS as well as the 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 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.

Step 4. List Current Settings

For DAG 3.5S 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.

Continued on next page

4.3 DAG 3.5S Card Capture Session, continued

Procedure, continued

Step 5. Check FPGA Image Loaded.

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

```
dag@endace:~$ dagld -d dag0 -x dag/xilinx/dag35pci-
erf.bit:dag/xilinx/dag35pp-erf.bit

dag@endace:~$ dagthree -d dag0
link    PoS noreset OC3c lt1 nodcr nolt0 fcl noeql
sonet   scramble slave
PoS     nopmax nopmin nocrc pscramble
packet  varlen slen=48
packetA drop=0
pci     33MHz 32-bit buf=32MB rxstreams=1 txstreams=0 mem=32:0
```

4.4 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:

Process Follow these steps configure the DAG 3.5S card in what you see can change style.

Step 1. Configure to ATM

Type:

```
dag@endace:~$ dagthree -d dag0 ATM
link    ATM noreset OC3c lt1 nodcr nolt0 nofcl noeql
sonet   scramble slave
ATM     ascramble noaidle noahec noapass noafix
packetA drop=0
pci     33MHz 32-bit buf=32MB rxstreams=1 txstreams=0 mem=32:0
```

Step 2. Configure for Other Options

For other options removing or adding the "no" prefix will change the setting:

```
dag@endace:~$ dagthree -d dag0 fcl
link    ATM noreset OC3c lt1 nodcr nolt0 fcl noeql
sonet   scramble slave
ATM     ascramble noaidle noahec noapass noafix
packetA drop=0
pci     33MHz 32-bit buf=32MB rxstreams=1 txstreams=0 mem=32:0
```

Continued on next page

4.5 DAG 3.5S Card Configuration Options

Description There are many DAG 3.5S card configuration options supported.

Options supported

`Atm`	set framer into ATM cell receive mode
`pos`	set framer into Packet-over-SONET (PoS) mode
`eth`	not supported
`raw`	not supported
`[no]reset`	hold/release framer [in] reset
`oc1c`	not supported
`oc3c`	set framer to OC3 receive mode
`oc12c`	set framer to 12c receive mode
`[no]lt1`	[un]set looptimer1. Do not touch
`[no]dcr`	[un]set disable clock recovery. Do not touch
`[no]lt0`	[un]set looptimer0. Do not touch
`[no]fcl`	[un]set facility loop back. This is useful for card chaining
`[no]eql`	[un]set equipment loop back. Do not touch
`[no]scramble`	[un]set SONET scrambling
`master`	generate SONET tx clock internally
`slave`	drive SONET rx clock from rx clock
`[no]pscramble`	[un]set Packet-over-SONET scrambling
`nocrc`	no PoS CRC checking
`crc16`	PoS CRC16 checks enabled
`crc32`	PoS CRC32 checks enabled
`[no]pmin`	dis/enable discard of packets smaller than a predefined minimum size
`[no]pmax`	dis/enable discard of packets larger than a predefined maximum size
`[no]afix`	when set correct single bit ATM HEC errors
`[no]apass`	not supported
`[no]ahec`	not supported
`[no]aidle`	when set pass through received idle cells
`[no]ascramble`	dis/enable descrambling of ATM cells. Keep set.

Values available only for variable length capture images are:

`slen=`	sets number of bytes of packet payload captured. Defaults to 48 for PoS, fixed at 52 for ATM.
`[no]varlen`	dis/enable variable length capture. Otherwise record length padded to slen. Defaults to varlen for PoS, fixed at novarlen for ATM.

Continued on next page

4.5 DAG 3.5S Card Configuration Options, continued

Inspect interface statistics

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

```
dag@endace:~$ dagthree -d dag0 -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`	Multiplexor loss of signal. If set, this indicates that there is either no signal at the receiver or the optical signal strength is too low to be recognized.
`bip3,bip2,bip1`	Bit interleaved parity byte error. These bits indicate a problem as reported by SONET B3, B2 and B1 overhead octets. If any of these bits are set, the card connection to the link is impaired. If oof and lof indicators are set along with bip's, the OCx carrier configuration is incorrect. Otherwise it indicates a signal problem related to either low light levels reaching the `dag` monitor, or true SONET-level errors as reported by SONET equipment operating the link to be monitored.
`lop`	Loss of pointer. If set the pointer processing logic has not locked to the SONET frame. It may indicate incorrect OC3c vs. OC12c setting.
`oof`	Out of frame. If set, the section overhead processor is not locked to the SONET stream. It may indicate incorrect OC3c vs. OC12c setting.
`lof`	Loss of frame. If set, oof had been asserted for more than 3 milliseconds.

Continued on next page

4.5 DAG 3.5S Card Configuration Options, continued

Status bits display, continued

`los` — Loss of signal
If set the framer has not detected any 0 to 1 transitions for 20 microseconds.

`label` — Path signal label.

Reflects the content of the SONET C2 overhead octet. Typical settings are:

13 ATM
16 PPP w/SPE scrambling
CF PPP wo/SPE scrambling

Changing values for this field indicate a SONET level error.

`lcd` — Loss of cell delineation.

If set the ATM state machine has no lock onto the ATM cell stream.

`sync` — ATM cell sync.

If set indicates the ATM cell engine has locked to ATM cell stream.

PoS OC3 stream example An example for a card locked to a PoS OC3c stream is:

```
los  bip3  bip2  bip1  lop  oof  lof  los   label
  0     0     0     0    0    0    0    0      Cf
  0     0     0     0    0    0    0    0      Cf
  0     0     0     0    0    0    0    0      Cf
  0     0     0     0    0    0    0    0      Cf
  0     0     0     0    0    0    0    0      cf
```

ATM< cell stream example An example for an ATM cell stream at OC12c is:

```
los  bip3  bip2  bip1  lop  oof  lof  los  label  lcd  Sync
  0     0     0     0    0    0    0    0     13    0     1
  0     0     0     0    0    0    0    0     13    0     1
  0     0     0     0    0    0    0    0     13    0     1
  0     0     0     0    0    0    0    0     13    0     1
  0     0     0     0    0    0    0    0     13    0     1
```

Continued on next page

4.5 DAG 3.5S Card Configuration Options, continued

Optical light levels

The following example is of a problem with optical light levels:

```
los   bip3   bip2   bip1   lop   oof   lof   los   label
  0      0      0      0     0     1     1     1      6d
  0      0      0      0     0     1     1     1      6d
  0      0      0      0     0     1     1     1      6d
  0      0      0      0     0     1     1     1      6d
  0      0      0      0     0     1     1     1      6d
```

The following example is of a card set to OC3 POS while the line carries OC12 POS.

```
los   bip3   bip2   bip1   lop   oof   lof   los   label
  0      0      0      0     0     1     1     1      d4
  0      0      0      0     0     1     1     1      90
  0      0      0      0     0     1     1     1      C9
  0      0      0      0     0     1     1     1      4a
  0      0      0      0     0     1     1     1      78
```

The following example is of a card set to OC3 POS while the line carries OC3 POS is:

```
los   bip3   bip2   bip1   lop   oof   lof   los   label
  0      0      0      0     0     1     1     0      08
  0      0      0      0     0     1     1     0      08
  0      0      0      0     0     1     1     0      08
  0      0      0      0     0     1     1     0      08
  0      0      0      0     0     1     1     0      08
```

No error bits are raised in `dagthree -si` if the card is configured to POS when on an ATM link or vice versa.

Network is ATM

If network is ATM, the:

- label should be 13
- lcd should be 0
- sync 1

Network is POS

If network is POS, the:

- label should be 16 for PPP or cf for HDLC
- lcd will be 1
- sync will be zero

It is still necessary to set the card mode correctly using `dagthree` in order to capture data!

4.6 Verify DAG 3.5S Card Configuration

Description The card configuration is verified as being correct by checking settings, path label for any errors.

Procedure Follow these steps to verify a card configuration.

Step 1. Check los Column

Ensure los (first column) is zero, and check light levels.

Step 2. Check settings

Ensure oof and lof are zero, otherwise change OC3c settings to OC12c or vice versa.

Step 3. Check for bip Errors

Ensure no bip errors occur, otherwise check cabling and light levels.

Step 4. Check Path Label

Ensure path label is correct as per the payload

Step 5. Check ATM Setting

Ensure ATM lcd is off and sync set.

Step 6. Check PoS Settings

Ensure PoS scrambling and CRC settings are correct.

4.6 General Purpose Counters

Description In addition to bit indicators, the card supports two general purpose counters which can be used to trouble-shoot network configuration problems.

The following items are countable.

0	sonet_bip1
1	sonet_bip2
2	sonet_bip3
3	atm_bad_hec
4	atm_cor_hec
5	atm_rcv_idle
6	atm_rcv_cell
7	pos_bad_crc
8	pos_min_err
9	pos_max_err
10	pos_abort
11	pos_good_frames
12	pos_bytes_rcvd

The item to be counted can be passed to `dagthree` with option `-c`.

4.7 Inspect Links Data and Cells

Description With ATM network configurations it is useful to inspect the number of data and idle cells on a link:

```
dag@endace:~$ dagthree -d dagN atm
dag@endace:~$ dagthree -d dagN -c 5,6 -i
```

For example, any given OC3c link the sum of data and idle cells per second should be around 350,000; 1.4 million for OC12c links respectively.

Trace file If tests provide a satisfactory status, a test trace is taken. A 10 seconds trace file is undertaken by passing option `-s 10` to `dagsnap`, along with option `-v` for more user information.

Description On Packet-over-SONET (PoS) links the number of good frames and data byte is a useful indicator to study, as shown below:

```
dag@endace:~$ dagthree -d dagN pos
dag@endace:~$ dagthree -d dag0 -c 11,12 -i
 pos_good_frames   pos_bytes_rcvd
              12              554
           70946          3825782
           71412          3850744
           71412          3850824
```

Continued on next page

4.7 Inspect Links Data and Cells, continued

Test trace

If the tests provide a satisfactory status, a test trace should be taken. We suggest a 10 seconds trace file by passing the option `-s 10` to `dagsnap`, along with option `-v` for more user information.

Incorrect scrambling settings

On Packet-over-SONET (PoS) links it can occur that very little or no data information is received. This typically indicates incorrect scrambling settings.

While a default is provided that matches typical link settings, the actual configuration varies from network to network.

To rectify incorrect scrambling settings, vary the scramble and pscramble options and retry.

4.8 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`.
8.	Firmware versions from `dagld -cx`.
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.

5.0 RUNNING DATA CAPTURE SOFTWARE

Introduction For a typical measurement session, the `scripts/dag35start` script is edited and used to operate the cards directly.

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

- Starting DAG 3.5S Card Capture Session
- High Load Performance

5.1 Starting DAG 3.5S Card Capture Session

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

For a typical measurement session, first move to the `dag` directory, load the driver, then load the appropriate Xilink receive image to each DAG. For example, for HDLC capture with two DAG 3.5S cards installed:

```
drv/dagload
tools/dagld –d dag0 –x xilinx/dag35pci-erf.bit:xilinx/dag35pp-erf.bit
tools/dagld –d dag1 –x xilinx/dag35pci-erf.bit:xilinx/dag35pp-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.

Process Follow this process to set a DAG card 3.5S card data capture session.

Process	Description

Start ATM capture session.	For ATM, start a capture session as follows:
	
	`dagthree –d dag0 atm`
	`dagsnap –v –o tracefile ncells=2 lcell`
	
	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.

Continued on next page

5.1 Starting DAG 3.5S Card Capture Session, continued

Process, continued

Process	Description
Capturing all ATM cells.	When `ncells=0`, all ATM cells are captured including OAM and RM cells. When `ncells` is non-zero, OAM and RM cells are not captured.
PoS capture session parameters.	For PoS, configure each card first: `dagthree –d dag0 pos` POS capture session parameters are set with dagthree. The card can operate in two modes, variable length capture (`varlen`), and fixed length capture (`novarlen`).
Variable length capture mode.	In variable length capture mode, a maximum capture size is set with slen=N bytes. This figure should be in the range 32 to 2048 and is rounded down to the nearest multiple of 4. Packets longer than slen will be truncated. Packets shorter than slen will produce shorter records, saving bandwidth and storage space. A full packet capture for example: dagthree –d dag0 varlen slen=1536

Continued on next page

5.1 Starting DAG 3.5S Card Capture Session, continued

Process, continued

Process	Description

Fixed length mode.	In fixed length mode, packets longer than the selected `slen` will be truncated to `slen`, but packets shorter than `slen` will produce records that are padded out to the value of `slen`.
	
	For this reason it is not recommended to use 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=48` (64 – ERF header size of 16):
	
	`dagthree -d dag0 novarlen slen=48`
	
	Capture setting must be set for each card in use.
	
	To start a capture session on a card, use `dagsnap`.
	
	`dagsnap -d dag0 -v -o tracefile0 &`
	`dagsnap -d dag1 -v -o tracefile1`

Stopping	By default `dagsnap` runs forever. `dagsnap` can be stopped with a signal: `killall dagsnap` `dagsnap` can also be configured to run for a fixed number of seconds and then exit using the `-s` option.

5.2 High Load Performance

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

Continued on next page

5.2 High Load Performance, continued

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.

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.

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]. The Endace ERF is detailed in Chapter 7 of this document.

Reading records

In order 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.

For OC-12/STM-4 (622Mbps) rates and above, 128MB or more may be required per card. To change the amount of memory reserved, edit the file `/etc/modules`. If the Endace Install CD has been used it will include this section:

```
# 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 must be less than 890MB.

6.0 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

6.1 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 (synchronize to
                           host clock)
         auxin             Aux input (unused)
         rs422out          Output the rs422 input signal
         loop              Output the selected input
         hostout           Output from host (unused)
         overout           Internal output (master card)
         set               Set DAG clock to PC clock
         reset             Full clock reset.  Load time
                           from PC, set rs422in, none out
```

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

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

6.2 Time Synchronization Configurations

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

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

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

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

6.2.1 Single Card no Reference Time Synchronization

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

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

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

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

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

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

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

6.2.2 Two Cards no Reference Time Synchronization

Description

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

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

Synchronizing cards

If both cards are to be accurately synchronized, 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

6.2.2 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.

6.2.3 Card with Reference Time Synchronization

Description

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 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

6.2.3 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 synchronizes to a valid signal.

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

Connecting time distribution server

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

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

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

Testing signal

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

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

```
dagpps -d dag0.
```

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

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

6.3 Synchronization Connector Pin-outs

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

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

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

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

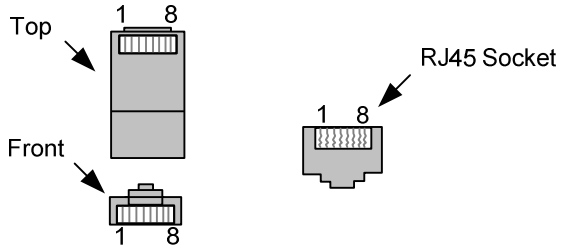

Figure 6-1. RJ45 Plug and Socket Connector Pin-outs.

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

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

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

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

7.0 DATA FORMATS OVERVIEW

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

- Data Formats
- Timestamps

7.1 Data Formats

Description The DAG card uses the ERF Type 1 HDLC PoS and Type 3 ATM cell records. 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. The diagram is not to scale.

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

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

7.1 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.

Figure Figure 7-2 shows the Type 1 PoS HDLC 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:1</td><td>flags</td><td>rlen</td></tr>
<tr><td colspan="2">lctr</td><td>wlen</td></tr>
<tr><td colspan="3">HDLC Header</td></tr>
<tr><td colspan="3">(rlen - 20) bytes of packet</td></tr>
</table>

Type 1 PoS HDLC Variable Length Record.

Continued on next page

7.1 Data Formats, continued

Table Table 7- 3 shows the Type 3 ATM Cell Record. The diagram is not to scale.

timestamp		
timestamp		
type:3	flags	rlen
lctr		wlen
ATM Header		
48 bytes of cell		

Table 7-3. Generic 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 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.

Continued on next page

7.2 Timestamps, continued

Example codes

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;
      }
```