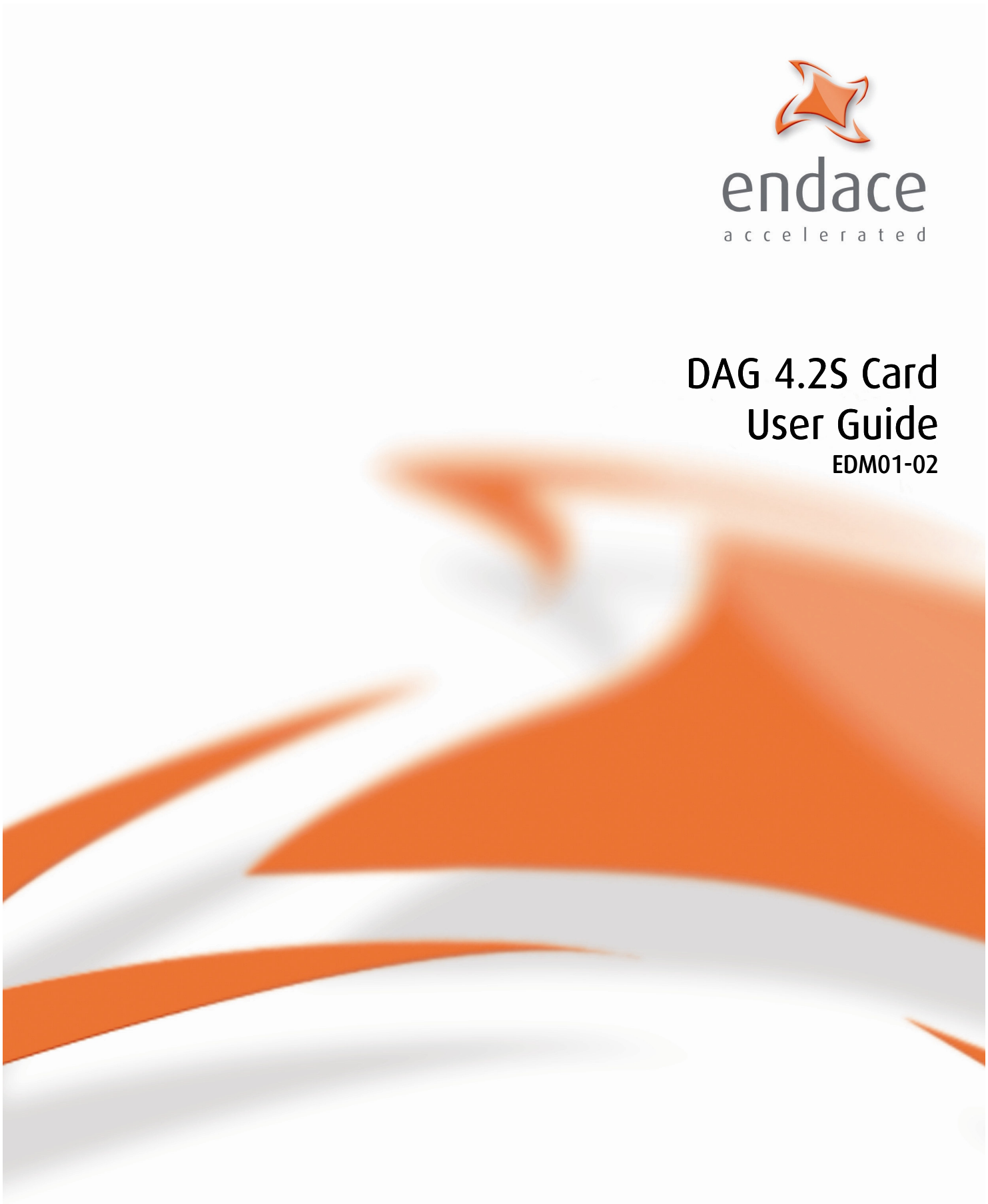

endace
accelerated

DAG 4.2S Card
User Guide
EDM01-02

www.endace.com

Published by:

Endace Measurement Systems® Ltd

Building 7
17 Lambie Drive

PO Box 76802
Manukau City 1702
New Zealand

Phone: +64 9 262 7260

Fax: +64 9 262 7261

support@endace.com

www.endace.com

International Locations

New Zealand
Endace Technology® Ltd

Level 9
85 Alexandra Street
PO Box 19246
Hamilton 2001
New Zealand

Phone: +64 7 839 0540
Fax: +64 7 839 0543

Americas
Endace USA® Ltd

Suite 220
11495 Sunset Hill Road
Reston
Virginia 20190
United States of America

Phone: ++1 703 382 0155
Fax: ++1 703 382 0155

Europe, Middle East & Africa
Endace Europe® Ltd

Sheraton House
Castle Park
Cambridge CB3 0AX
United Kingdom

Phone: ++44 1223 370 176
Fax: ++44 1223 370 040

Protection Against Harmful Interference

When present on equipment this manual pertains to, the statement "This device complies with part 15 of the FCC rules" specifies the equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the Federal Communications Commission [FCC] Rules.
These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment.
This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications.
Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

Extra Components and Materials

The product that this manual pertains to may include extra components and materials that are not essential to its basic operation, but are necessary to ensure compliance to the product standards required by the United States Federal Communications Commission, and the European EMC Directive. Modification or removal of these components and/or materials, is liable to cause non compliance to these standards, and in doing so invalidate the user's right to operate this equipment in a Class A industrial environment.

Disclaimer

Whilst every effort has been made to ensure accuracy, neither Endace Measurement Systems Limited nor any employee of the company, shall be liable on any ground whatsoever to any party in respect of decisions or actions they may make as a result of using this information.

Endace Measurement Systems Limited has taken great effort to verify the accuracy of this manual, but assumes no responsibility for any technical inaccuracies or typographical errors.

In accordance with the Endace Measurement Systems policy of continuing development, design and specifications are subject to change without notice.

Table of Contents

Chapter 1: Introduction

Introduction The installation of the Endace DAG card on a PC begins with installing the operating system and the Endace software.

Viewing this document This document, DAG 4.2S Card User Manual is available when the installation CD is placed in a running Windows PC.

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

1.1 User Manual Purpose

Description The purpose of this DAG Card User Manual is to describe:
Installing DAG 4.2S card
Setting Optical Power
DAG 4.2S Card Confidence Testing
Running Data Capture Software
Synchronizing Clock Time
Data Formats

Prerequisite This document presumes the DAG card is being installed in a PC already configured with an operating system.

A copy of 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 EDM04.06-01 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 EDM04.06-02 Windows Installation Manual, packaged in the CD shipped with the DAG card.

1.2 DAG 4.2S Card Description

Description The DAG 4.2S single interface OC-48c/STM-16c card is capable of cell and packet capture and generation on IP networks.

Description Figure 1-1 shows the DAG 4.2S PCI card.

Figure 1-1. DAG 4.2S PCI Card.

1.3 DAG 4.2S Card Architecture

Description Serial SONET optical data is received by the DAG 4.2S card optical interface, and fed through a demultiplexor into a physical layer ASIC. The packet data is then fed immediately into the Xilinx FPGA. This FPGA contains the DUCK timestamp engine, packet record processor, and PCI interface logic.

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 an external FIFO before transmission to the host.

Figure

Figure 1-2 shows the DAG 4.2S card major components and data flow.

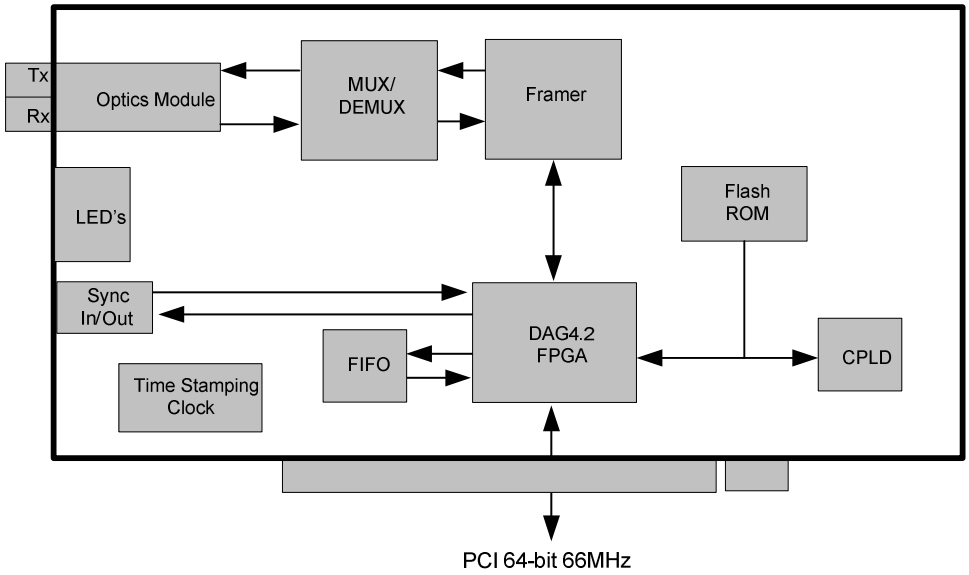

Figure 1-2. DAG 4.2S Card Major Components and Data Flows.

1.5 DAG 4.2S Card System Requirements

Description

The DAG 4.2S card and associated data capture system minimum operating requirements are:

- PC, at least Pentium III 800MHz or faster
- Intel i840, ServerWorks III LE/HE or newer chip set
- Minimum of 128 MB RAM
- At least one free 64-bit 3.3v signaling only PCI slot 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

Chapter 2: Installation

Introduction A DAG 4.2S card can be installed in any free 3.3v signalling 64-bit Bus Mastering PCI slot. The DAG 4.2S card is capable of running at 66MHz, but if any other device on the same bus is not capable of 66MHz operation then all devices on the bus will operate at 33MHz.

Although by default, the driver supports up to four DAG cards in one system, there should not be more than two cards on a single PCI bus due to bandwidth limitations. However, 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.

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 on Linux/FreeBSD operating systems by following the instructions in EDM04-01 Linux/FreeBSD Installation Guide.

To install the software on a Windows operating system, follow the instructions in EDM04-02 Windows Installation Guide.

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

2.2 Insert the Card into PC

Description Inserting the DAG 4.2S card into a PC involves accessing the bus slot, fitting the card, and replacing bus slot screw.

Procedure Follow these steps to insert the DAG 4.2S card.

Step 1. Access bus Slot

Power computer down.

Remove PCI bus slot cover.

Procedure (continued)

Step 2. **Fit Card**

Insert into PCI bus slot.

Step 3. **Replace bus Slot Screw**

Secure card with screw.

Step 4. **Power up Computer**

2.3 DAG 4.2S Card Optical Connectors

Description The DAG 4.2S card has two SC-type optical connectors. The bottom connector nearest the PCI slot is for the received signal, the top is for the transmitted signal.

The transmit port is only connected if the loop back facility being used in the DAG to daisy chain systems, or if a data generation program being used.

If the Tx port of the DAG 4.2S card is not used, the SC-type transceiver optics should be covered to prevent ingress of dust.

An 8-pin RJ45 socket is used for time synchronization. This socket should never be connected to an Ethernet network or telephone line.

Figure Figure 2-1 shows the DAG 4.2S card SC-type optical connectors.

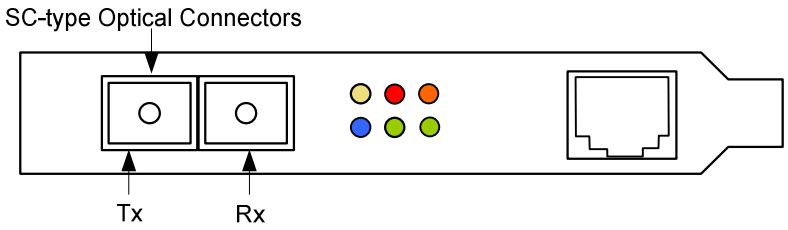

Figure 2-1. DAG 4.2S Card SC-type Optical Connectors.

Chapter 3: Setting Optical Power

Introduction

The optical power range depends on the particular device fitted on the DAG 4.2S card.

The DAG 4.2S card is shipped fitted with HFCT 5402D 1300nm single-mode short range optics 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 4.2S card optics power module part, single-mode fibre [SMF], and configuration.

Part No.	Fibre	Data Rate	Max Pwr [dBm]	Min Pwr [dBm]	Nominal pwr [dBm]
HFCT 5402D	SMF	2488	-3	-18.5	-14

MMF: Multi-Mode Fibre SMF: Single-Mode Fibre

In this chapter

This chapter covers the following sections of information.

3.1 DAG 4.2S Card Optical Power Input

Description

The optical power input to the DAG 4.2S card must be within a receiver's dynamic range.

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.

Input power

When the DAG card is set up, measure the optical power at the receiver and ensure that it is within the specified power range. The recommended power is -14 dBm.

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 dBm and 4 dBm on each output

90:10 splitter will have losses of about 10 dBm in the high loss output, and <2 dBm 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.

Wavelength loss

Splitters are designed for a particular wavelength. When mismatched, the split ratio will be different from that which was intended.

Chapter 4: Confidence Testing

Introduction The confidence testing is a process to determine whether the DAG 4.2S 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.

4.1 Interpreting DAG Card LED Status

Description The DAG 4.2S card has a block of 6 status LEDs, one blue, one yellow, two green, one red and one orange.

Figure Figure 4-2 shows the DAG 4.2S card status LEDs.

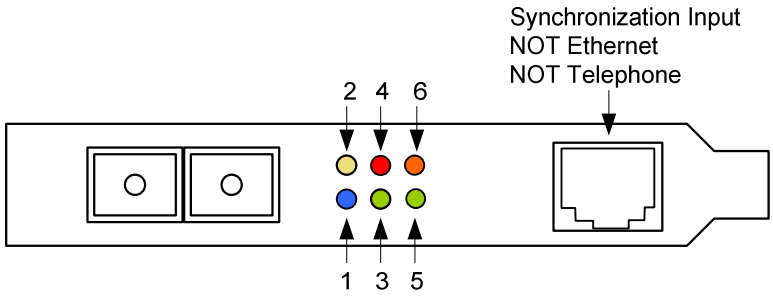

Figure 4-2. DAG 4.2S Card Status LEDs.

LED definitions The following table describes the LED definitions.

LED	Description
LED 1 - Blue	FPGA successfully programmed.
LED 2 - Yellow	Data capture in progress.
LED 3 - Green	PPS synchronisation signal, flashes with valid input signal.
LED 4 - Red	Transmitter laser ON.
LED 5 - Green	SD. Signal Detect, valid optical signal seen by the optical receiver.
LED 6 - Orange	LOF. Loss of Frame synchronization alarm, usually caused by loss of signal.

Description When the DAG 4.2S card is powered up for a capture session the top left LED 1 should always come on, and:

- LED 2 indicates when a packet capture session is in progress.
- LED 3 flashes if a PPS signal is being received by the card.
- LED 4 is only on if the laser is turned on with the `dagfour` utility. When an OC-48c optical signal is applied.
- LED 5 should go on.
- LED 6 should go out.

Figure Figure 4-3 shows the correct LED state for the DAG 4.2S card without optical input.

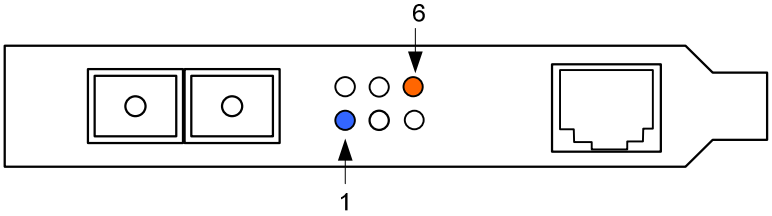

Figure 4-3. DAG 4.2S Card Correct LED Status Without Optical Input.

Description The `dagfour` utility supports configuration status and physical layer interface statistics for the DAG 4.2S card.

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 (`dagfour -h`) as well as via the manual page.

4.2 DAG 4.2S Card Capture Session

Description The DAG 4.2S card uses a VSC9112 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 OC-48c and STM-16c standards.

A successful DAG 4.2S 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 DAG for normal use.

Procedure Follow these steps to troubleshoot DAG 4.2S card configuration.

Step 1. Check Receiver Ports Optical Signal Levels.

The card supports 1300 nanometer single-mode fibre attachments with optical signal strength between 0 dBm and -18 dBm.

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

The card receiver ports are the lower half of the dual SC-style connector, the closest to the LED's.

Cover the unused card transmit port with an SC-style plug 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 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 4.2S framer configuration and statistics the `dagfour` tool is supplied.

Calling `dagfour` without arguments lists current settings.

`The dagfour -h` prints a help listing on tool usage.

Step 5. Check FPGA Image Loaded.

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

Procedure, continued

```
dag@endace:~$ dagrom -rvp -d dag0 -f xilinx/dag423pcix-erf.bit

dag@endace:~$ dagfour -d dag0
light   nolaser nomuxfcl nomuxeql
link    PoS noreset OC48c nofcl noeql
sonet   slave scramble
PoS     nopmin nopmax nodiscard crc32 pscramble norxpkts notxpkts
long=1500 short=40
packet  varlen slen=48
packetA drop=0
pci     66MHz 64-bit buf=128Mb rxstreams=1 txstreams=0 mem=128:0
```

Step 6. Configure DAG for Normal Use

The `dagfour default` command is always used:

```
dag@endace:~$ dagfour default
light   nolaser nomuxfcl nomuxeql
link    PoS noreset OC48c nofcl noeql
sonet   slave scramble
PoS     nopmin nopmax nodiscard crc32 pscramble norxpkts notxpkts
txrclk=78MHz long=61440 short=40
packet  varlen slen=48
packetA drop=0
pci     66MHz 64-bit buf=128Mb rxstreams=1 txstreams=0 mem=128:0
```

4.3 DAG 4.2S Card Configuration in WYSYCC Style

Description The configuration of the tool works in WYSIWYC style – what you see is what you can change.

To turn on the card's laser for instance, type:

```
dag@endace:~$ dagfour -d dag0 laser
light   laser nomuxfcl nomuxeql
link    PoS noreset OC48c nofcl noeql
sonet   slave scramble
PoS     nopmin nopmax nodiscard crc32 pscramble rxpkts txpkts
txrclk=78MHz long=61440 short=40
packet  varlen slen=48
packetA drop=0
pci    66MHz 64-bit buf=128Mb rxstreams=1 txstreams=0 mem=128:0
```

ATM status In ATM mode, the status output changes:

```
dag@endace:~$ dagfour -d dag0 atm
light   laser nomuxfcl nomuxeql
link    ATM noreset OC48c fcl noeql
sonet   slave scramble
packetA drop=0
pci     66MHz 64-bit buf=128Mb rxstreams=1 txstreams=0 mem=128:0
```

For configuration options removing or adding the `no` prefix will change the setting.

4.4 DAG 4.2S Card Configuration Options

Description There are many DAG 4.2S card configuration options supported.

`default`	set framer to normal defaults
`[no]laser`	dis/enable transmit laser
`atm`	set framer into ATM mode
`pos`	set framer into Packet-over-SONET (PoS) mode
`[no]muxfcl`	(un)set facility loopback in the MUX
`[no]muxeql`	(un)set equipment loopback in the MUX
`[no]reset`	hold/release framer (in) reset
`oc48c`	set framer to OC48c mode
`[no]fcl`	(un)set facility loop back in the phy. This is useful for card chaining
`[no]eql`	(un)set equipment loop back in the phy
`[no]scramble`	(un)set SONET scrambling
`master`	set card to SONET clock master
`slave`	set card to SONET clock slave
`[no]pscramble`	(un)set Packet-over-SONET scrambling
`[non]discard`	(un)set discard of FCS mismatched PoS packets
`crc16`	PoS CRC16 link
`crc32`	PoS CRC32 link
`[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]rxpkts`	dis/enable packet reception
`[no]txpkts`	dis/enable packet transmission
`[no]txrclk`	dis/enable packet transmission

`slen=X`	capture X bytes of packet data.
`[no]varlen`	dis/enable variable length capture. Otherwise record length padded to slen. Defaults to varlen.

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:~$ dagfour -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 the following example, the interval is set to one second via the `-i` option.

`LoS`	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.
`OoF`	Out of frame. If set, the section overhead processor is not locked to the SONET stream.
`LoF`	Loss of frame. If set, `Oof` had been asserted for more than 3 milliseconds.
`SectionBIP`	SONET/SDH Section Bit Interleaved Parity error. The link is impaired, check connections and optical signal level.
`LineBIP`	SONET/SDH Line Bit Interleaved Parity error. The link is impaired, check connections and optical signal level.
`LineFEBE`	SONET/SDH Line Far End Bit Error. The link is impaired, check connections and optical signal level.
`PathBIP`	SONET/SDH Path Bit Interleaved Parity error. The link is impaired, check connections and optical signal level.
`PathFEBE`	SONET/SDH Path Far End Bit Error. The link is impaired, check connections and optical signal level.

POS mode In POS mode, the following columns are present:

`RxFrames`	Number of PoS frames received since last reading.
`RxBytes`	Number of PoS payload bytes received since last reading.

ATM mode In ATM mode, the following columns are present:

`LCD`	Loss of Cell Delineation. The framer cannot lock onto the ATM cells.
`RxCells`	Number of non-idle ATM cells received since last reading.

Extra counters These extra counters are available with the extended statistics option:

```
dagfour –ei
```

`Abort`	Number of PoS frames aborted since last reading.
`FCSErr`	Number of PoS frames with FCS errors since last reading.
`Invalid`	Number of Invalid PoS frames received since last reading.
`Path_Label`	SONET/SDH C2 byte value or Path Signal Label. Typically 0x13 for ATM, 0x16 for PPP, and 0xCF for Cisco HDLC POS.

PoS OC-48 stream example An example for a card locked to a PoS OC-48c stream carrying a constant traffic load is:

LoS	OoF	LoF	SectionBIP	LineBIP	LineFEBE	PathBIP	PathFEBE	RxFrames	RxBytes
0	0	0	335492	56699364	9083393	619149	397	22030639	4294901759
0	0	0	0	0	0	0	0	150316	5712008
0	0	0	0	0	0	0	0	151025	5738950
0	0	0	0	0	0	0	0	151026	5738988

The first second has high values as the counters have accumulated their values over more than one second.

Optical light level problem The following situation indicates a problem with the optical light levels:

LoS	OoF	LoF	SectionBIP	LineBIP	LineFEBE	PathBIP	PathFEBE	RxFrames	RxBytes
0	0	0	3943886	769018395	126206289	7887612	174	472777619	4294901759
1	1	1	8048	1569360	257504	16096	0	0	298988352
1	1	1	8095	1578525	259040	16190	0	0	300750950
1	1	1	8096	1578720	259072	16192	0	0	300780107
1	1	1	8097	1578915	259104	16192	0	0	300810210
1	1	1	8093	1577940	258976	16186	0	0	300659848

Stabilise configuration Follow these steps to stabilise the configuration.

Step 1. Ensure Columns are at Zero

Check that the `LoS`, `OoF`, and `LoF`, being the first three columns, are zero.

Check light levels.

Step 2. Inspect for BIP Errors

Check that no BIP errors occur, otherwise check cabling and light levels.

Step 3 Check CRC Settings

For PoS, ensure scrambling and CRC settings are correct.

Step 4. Check FEBE Errors

FEBE errors indicate that the remote end of the link is detecting errors.

NOTE: This may not affect the capture of data by the DAG card.

4.5 Inspect Links Data and Cells

Description On Packet-over-SONET (PoS) links it can happen that there is very little or no data information 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.

A remedial action is to vary the scramble and pscramble options and performing a retry.

Description, continued

If it is necessary to connect the transmit port of the DAG 4.2S card to other equipment, it is necessary to enable the transmit laser. The laser normally used is eye safe, but is disabled as a precaution as it is not normally needed.

The laser radiation is in the invisible infrared part of the spectrum. When the laser is turned on, the red laser warning LED will be lit.

In a test-bench situation where two DAG cards are connected directly to each other, one card must be designated the SONET clock master. This can only be done on cards fitted with 77MHz SONET master clock oscillator crystal.

In normal use the DAG card should be the SONET clock slave, deriving its signals from the received network stream.

4.6 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.	UNIX operating system only. Any compiler errors or warnings when building DAG driver or tools.
6.	UNIX operating system only. 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.	UNIX operating system only. Output of `cat /proc/dag`.
8.	Firmware versions from `dagbug -cx` and `dagrom -x`.
9.	Physical layer status reported by: `dagfour`
10.	Network link statistics reported by: `dagfour -ei`
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.

Chapter 5: Running Data Capture Software

Introduction For a typical measurement session, the `scripts/dag42start` script is edited and used to operate the cards directly.

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

5.1 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 Setting a data capture session is described in the following process.

Process	Description
Load Xilink receive image.	For a typical measurement session, first move to the `dag` directory. Load the driver. Load the Xilinx receive image to each DAG. For example, with two DAG 4.2S cards installed:

`drv/dagload`
`tools/dagrom -rvp -d dag0 -f xilinx/dag423pos-erf.bit`
`tools/dagrom -rvp -d dag1 -f xilinx/dag423pos-erf.bit`	

	Although the images are named pos, they can be configured to capture ATM traffic.

Check integrity.	Set, and then check the integrity of the physical layer to both DAGs.
	
	`tools/dagfour -d dag0 default`
	`tools/dagfour -d dag1 default`

Process, continued

Process	Description
Setting capture session parameters.	Parameters are set with `dagfour`. 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 2048 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/dagfour -d dag0 varlen slen=1536`
Setting fixed length mode.	In fixed length mode, packets longer than the selected `slen` are truncated to `slen`. Packets shorter than slen produce records padded out to `slen` length. Large `slen` values in fixed length mode should not be used because short packets arriving produce large padded records, wasting bandwidth and storage space. An example, for fixed length 64-byte records, choose `slen=48` (64 – ERF header size of 16): is: `tools/dagfour -d dag0 novarlen slen=48`

Process, continued

Process	Description

Starting a capture session.	Once the capture parameters are configured, a capture session is started by:
	
	`tools/dagsnap -d dag0 -v -o tracefile0 &`
	`tools/dagsnap -d dag1 -v -o tracefile1`
	
	Option `-v` provides user information during capture; it can be omitted for automated trace runs.
	
	If the `tracefile` parameter is not specified the tool writes to `stdout`, which can be used to pipeline `dagsnap` with other tools from `dagtools` package.
	
	By default `dagsnap` runs forever.

Stopping	`dagsnap` can be stopped with a signal: `killall dagsnap`, or key strokes CTL + C `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.

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 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 will be discarded by the DAG card. Any loss can be detected in-band by observing the `lctr` [Loss Counter] field of the Extensible Record Format.

Avoiding packet loss

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. Capture at OC-12/STM-4 (622Mbps) rates and above may require a larger buffer.

128MB or more may be required per card.

The amount of memory reserved is changed by editing 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.

Chapter 6: Synchronizing Clock Time

Description

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

The system that provides the DAG synchronisation capability is known as the DAG Universal Clock Kit (DUCK).

An independent clock in each DAG card runs from the PC clock. A card's clock is initialised using the PC clock, and then free-runs using a crystal oscillator.

Each card's clock can vary relative to a PC clock, or other DAG cards.

DUCK configuration

The DUCK is configured to avoid time variance between sets of DAG cards or between DAG cards and coordinated universal time [UTC].

Accurate time reference can be obtained from an external clock by connecting to the DAG card using the synchronisation connector, or the host PCs clock can be used in software as a reference source without additional hardware.

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

Common synchronization

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

Common synchronisation sources include GPS or CDMA (Cellular telephone) time receivers.

Endace produces the TDS 2 Time Distribution Server modules and the TDS 6 units that enable multiple DAG cards to be connected to a single GPS or CDMA unit.

More information is 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.

6.1 Configurations 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 synchronisation from several input sources, and can 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.

```
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
```

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.

6.2.1 Single Card no Reference Time Synchronization

Description When a single card is used with no external reference, the card can be synchronised 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 synchronisation 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  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 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 synchronise to the host and also act as a master for the second.

Synchronizing 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 clocks 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.

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 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 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 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 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 synchronisation port should never be connected to Ethernet or telephone equipment.

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

Testing 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.

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.

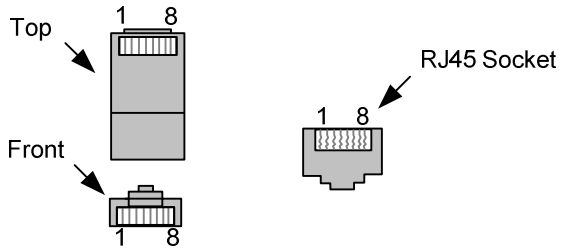

Figure 6-1. RJ45 Plug and Socket Connector Pin-outs.

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 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.

Chapter 7:Data Formats

Introduction The DAG 4.2S card supports a record format known as the Extensible Record Format [ERF].

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

Data Formats
Timestamps

7.1 Data Formats

Description The DAG 4.2S card uses the ERF Types 1 and 3 timestamps. 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 Type 1 PoS HDLC record.

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

Table 7-1. Type 1 PoS HDLC Record.

Data format The following is a description of the Type 1 PoS HDLC record field.

Field	Description
HDLC Header	Length may vary depending on protocol.

Table Table 7-2 shows the Type 2 Ethernet record.

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

Table 7-2. Type 2 Ethernet Record.

Data format

The following is a description of the Type 2 Ethernet record offset field.

Field	Description
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.

Data format overview

The following is an overview of the data formats 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 record 7: `TYPE_MC_ATM`: Multi-channel ATM Cell 8: `TYPE_MC_RAW_CHANNEL`: Multi-channel raw link data. 9: `TYPE_MC_AAL5`: Multi-channel AAL5 frame 10: `TYPE_COLOR_HDLC_POS`: PoS with HDLC framing and packet classification information. 11: `TYPE_COLOR_ETH`: Ethernet with packet classification information.

Data format overview, continued

Data Format	Description
flags	This byte is divided into 2 parts, the interface identifier, and a set of 1-bit flags. 1-0: Capture interface 0-3. 2: Varying record lengths present. 3: Truncated record [insufficient buffer space]. 4: rx error [link error]. 5: ds error [internal error]. 6-7: 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`. NOTE: This is not present in Type 10 and Type 11 ERF records.
wlen: *wire length*	Packet length including some protocol overhead. The exact interpretation of this quantity depends on the 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.

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 is also available to Windows users in the Winsock library.

Different DAG cards have different actual resolutions. This is accommodated by the lowermost bits that are not active being set to zero. In this way the interpretation of the timestamp does not need to change when higher resolution clock hardware is available.

Example code

Here is some example code showing how a 64-bit ERF timestamp (erfts) can be converted into UNIX 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;
}
```