DAG 3.8S Card User Manual

2.5.5r1

EDM01.05-09r1

Endace Measurement Systems | New Zealand | www.endace.com

Leading Network Intelligence

Published by:

Endace Measurement Systems® Ltd
Building 7
17 Lambie Drive
PO Box 76802
Manukau City 1702
New Zealand
Phone: +64 9 262 7260
Fax: +64 9 262 7261
support@endace.com
www.endace.com

International Locations

New Zealand

Endace Technology® Ltd
Level 9
85 Alexandra Street
PO Box 19246
Hamilton 2001
New Zealand
Phone: +64 7 839 0540
Fax: +64 7 839 0543
support@endace.com
www.endace.com

Americas

Endace USA® Ltd
Suite 220
11495 Sunset Hill Road
Reston
Virginia 20190
United States of America
Phone: ++1 703 382 0155
Fax: ++1 703 382 0155
support@endace.com
www.endace.com

Europe, Middle East & Africa

Endace Europe® Ltd
Sheraton House
Castle Park
Cambridge CB3 0AX
United Kingdom
Phone: ++44 1223 370 176
Fax: ++44 1223 370 040
support@endace.com
www.endace.com

Typographical Conventions Used in this Document

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

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

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

Protection Against Harmful Interference

When present on product this manual pertains to and indicated by product labelling, the statement "This device complies with part 15 of the FCC rules" specifies the equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the Federal Communications Commission [FCC] Rules.

These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment.

This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications.

Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

Extra Components and Materials

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

Table of Contents

USE THIS SPACE FOR NOTES

1.0 PREFACE

Introduction

The installation of the Endace DAG 3.8S 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 3.8S Card User Manual is available on the installation CD.

In this chapter

This chapter covers the following sections of information.

- User Manual Purpose
- DAG 3.8S Card Product Description
- DAG 3.8S Card Architecture
- DAG 3.8S Card Extended Functions
- DAG 3.8S Card System Requirements

1.1 User Manual Purpose

Description

The purpose of this DAG 3.8S Card User Manual is to describe:

- Installing DAG 3.8S Card
- Setting DAG 3.8S Card Optical Power
- Confidence Testing DAG 3.8S 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 3.8S Card Product Description

Description

The DAG 3.8S card standard configuration supports receive-only packet and cell capture applications for OC3c or OC12c, ATM or Packet-over-SONET (PoS) networks.

The DAG3.8S card has two transceivers which can be operated simultaneously allowing a single card to monitor one or both directions of a full-duplex link.

The DAG 3.8S card is also capable of transmitting packets at 100% line rate on both ports while simultaneously receiving packets at 100% line rate on both ports.

Figure

Figure 1-1 shows the DAG 3.8S PCI-X card.

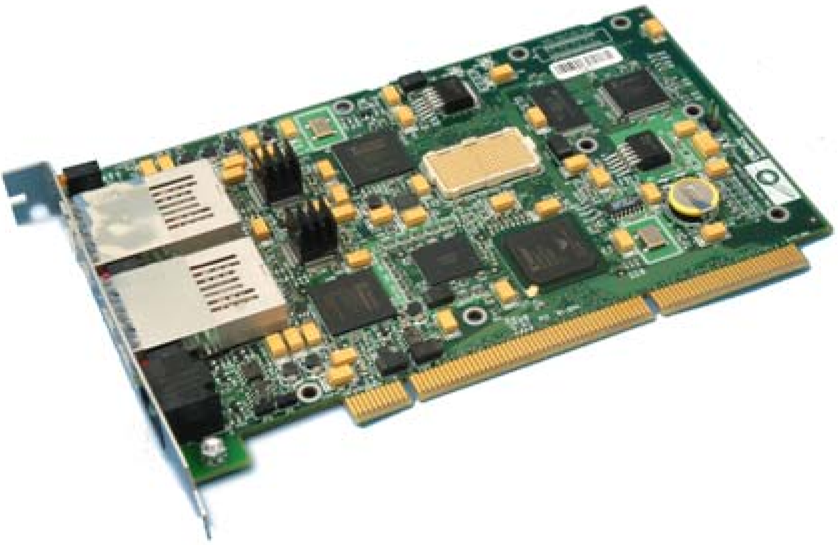

Figure 1-1. DAG 3.8S PCI-X Card.

1.3 DAG 3.8S Card Architecture

Description Serial SONET/SDH optical data is received by two optical interfaces, and passed through demultiplexors.

The network data feeds immediately into two physical layer FPGAs. The SONET/SDH payload data is then sent to the main FPGA.

This FPGA contains the DUCK timestamp engine, packet record processor, and PCI-X interface logic.

Because of component close association, packets or cells are time-stamped accurately. Time stamped packet records are stored in an external FIFO before transmission to the host.

Figure Figure 1-2 shows the DAG 3.8S card major components and data flow.

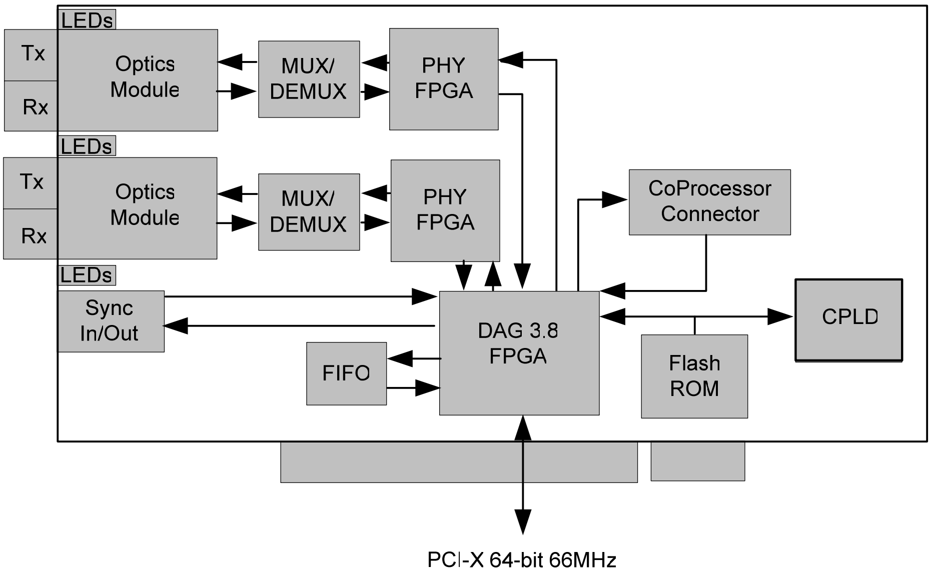

Figure 1-2. DAG 3.8S Card Major Components and Data Flow.

1.4 DAG 3.8S Card Extended Functions

Description The functionality of the DAG 3.8S can can be extended in many ways. The framers are normally set up to map STM-1, STS-3c, STM-4c and STS-12c payloads, but other mappings are possible.

The DAG 3.8S card is equipped with a coprocessor connector which can be used with the optional Endace DAG Coprocessor as a data processing tool.

Continued on next page

1.4 DAG 3.8S Card Extended Functions, continued

Coprocessor IP filtering/packet classification

The specifications for the coprocessor IP filtering/packet classification are:

- Packets are classified by TCP/IP header fields and/or payload content.
- Up to 16,384 TCP/IP header classification rules.
- Classification rules are assigned a user-defined 16-bit identifier
- Packets matching classification rules are assigned the matching rule's identifier.
- Programmable actions may be associated with each rule identifier. For example, The packet should either be dropped, or presented to the host.
- Packets presented to the host include the rule-match identifier in the record header.

Coprocessor ATM AAL5 Reassembly

The DAG 3.8S card is equipped with a coprocessor which can be used as a data processing tool with the optional Endace DAG Coprocessor.

The ATM AAL5 Reassembler specifications are:

- Supports up to 65,535 simultaneously active VCI/VPI's
- Supports simultaneous reassembly of up to 65,536 AAL5 frames
- Selected VPI/VCI discovery, state management, and ageing
- Supports up to full OC-12/STM-4 cell rate on both interfaces simultaneously [~2.8 Mcells/sec]
- Rich statistics, counters, and timeouts

Contact Endace support

Contact the Endace customer support team at support@endace.com for:

- DAG coprocessor pricing details
- Purchase of a DAG coprocessor
- Information on enabling effective use of extended functions

1.5 DAG 3.8S Card System Requirements

Description

The DAG 3.8S card and its associated data capture system has the following minimum system requirements:

- PC, at least Intel Xeon 1.8GHz or faster
- Intel E7500, ServerWorks Grand Champion LE/HE or newer chip set
- Minimum of 256 MB RAM
- At least one free PCI-X 1.0 slot supporting 66MHz operation
- Software distribution requires 30MB free space
- 6GB for installation of Endace software, which is optional

Continued on next page

1.5 DAG 3.8S Card System Requirements, continued

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

USE THIS SPACE FOR NOTES

2.0 INSTALLING DAG 3.8S CARD

Description

A DAG 3.8S card can be installed in any free PCI-X 1.0 slot.

The DAG 3.8S card operates only in 66MHz PCI-X mode. If placed into a slot rated for higher speeds the bus will automatically change to 66MHz, including any other devices sharing the bus.

The DAG 3.8S should be the only device on the PCI-X bus if possible. The DAG 3.8S will not operate in 32 or 64-bit PCI slots.

In this chapter

This chapter covers the following sections of information.

- Installation of Operating System and Endace Software
- Insert DAG 3.8S Card into PC
- DAG 3.8S Card Port Connectors

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 3.8S Card into PC

Description

Inserting the DAG 3.8S 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 3.8S card.

Step 1. **Access bus Slot**

Power computer down.

Remove PCI-X bus slot cover.

Step 2. **Fit Card**

Insert into PCI-X bus slot.

Step 3. **Replace bus Slot Screw**

Secure card with screw.

Step 4. **Power up Computer**

Continued on next page

2.3 DAG 3.8S Card Port Connectors

Description

There are two SC-type optical port connectors. One pair for signals transmitting, the second pair is for receiving signals.

Each port upper connector consists of an optical fibre transmitter and receiver.

The upper connector of each pair is for transmitting signals. They are connected only if loop-back facility is used in the DAG card to daisy-chain the systems. They are also connected if a data generation program is being used.

The bottom connectors of each pair is used for the received signal.

Part/function

The port functions are described in the following table.

Part	Function
8-pin RJ45 socket.	Used for time synchronization input. CAUTION: This socket should never be connected to an Ethernet network or telepphone line.

2.4 DAG 3.8S Card Pluggable Optical Transceivers

Description

Some newer versions of the DAG 3.8S cards are available with pluggable optics. To provide compatibility with the broadest possible range of optical parameters, Endace offers the industry standard Small Form-factor Pluggable [SFP] optical transceiver on the DAG 3.8S card.

The SFP transceiver consists of two parts:

- Mechanical chassis attached to the circuit board
- Transceiver unit which may be inserted into the chassis

The correct transceiver is chosen to suit the optical parameters of the target network installed in the chassis.

The transceiver may then be connected to the network via LC-style optical connectors.

Further information about the Pluggable Optical Transceiver is available at the Endace http://www.endace.com/dagPluggable.htm web page.

Continued on next page

2.4 DAG 3.8S Card Pluggable Optical Transceivers, continued

Figure

Figure 2-1 shows the pluggable optical transceivers.

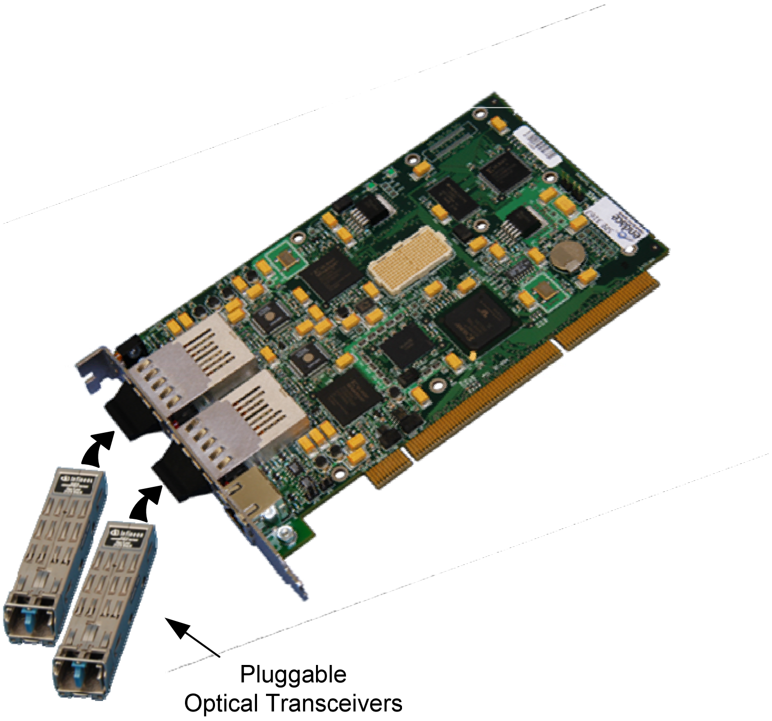

Figure 2-1. Pluggable Optical Transceivers.

USE THIS SPACE FOR NOTES

3.0 SETTING DAG 3.8S CARD OPTICAL POWER

Description

The optical power range depends on the particular transceiver module fitted to the DAG 3.8S card.

The power range depends on the particular device installed on the DAG card. The DAG 3.8S card is shipped fitted with HFBR 5208EM 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 µW.

Configuration

The following table describes the DAG 3.8S card optics power module configuration. MMF = Multi Mode Fibre. SMF = Single Mode Fibre.

Part #	Fibre	Data Rate	Max Power [dBm]	Min Power [dBm]	Nominal Pwr [dBm]
HFBR 5208EM	MMF	155/622	-14	-26	-20
HFCT 5208EM	SMF	155/622	-7	-28	-20

In this chapter

This chapter covers the following sections of information.

- DAG 3.8S Card Optical Power Input
- Splitter Losses

3.1 DAG 3.8S Card Optical Power Input

Description

The optical power input into the DAG 3.8S card must be within the 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.

Continued on next page

3.1 DAG 3.8S Card Optical Power Input, continued

Input power

The DAG 3.8S 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 3.8S CARD

Introduction

The confidence testing is a process to determine the DAG 3.8S 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 3.8S Card LED Status
- DAG 3.8S Card Configuration
- Configuration in WYSYCC Style
- DAG 3.8S Card Configuration Options
- Verify DAG 3.8S Card Configuration
- General Purpose Counters
- Inspect Links Data and Cells
- Reporting Problems

4.1 Interpreting DAG 3.8S Card LED Status

Description

The DAG 3.8S has a 8 status LEDs, one blue, three green, two orange, and two red.

When a DAG 3.8S series card is powered up the LED 1 should always come on indicating the FPGA is successfully programmed.

The LED 4 and LED 6 are on when the Loss of Pointer or Loss of Framing conditions are true, or if Loss of Cell Delineation is true only in ATM mode.

Continued on next page

4.1 Interpreting DAG 3.8S Card LED Status, continued

Figure

Figure 4-1 shows the DAG 3.8S status LEDs.

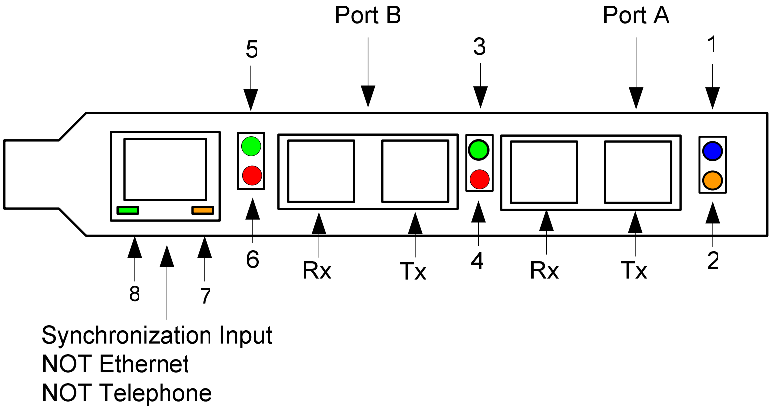

Figure 4-1. DAG 3.8S Card Status LEDs.

LED definitions

The following table describes LED display definitions.

LED	Display Function
LED 1	FPGA successfully programmed.
LED 2	Data capture in progress.
LED 3	Port A Signal Detect – valid optical signal seen by the optical receiver.
LED 4	Port A Link Error.
LED 5	Port B Signal Detect – valid optical signal seen by the optical receiver.
LED 6	Port B Link Error.
LED 7	PPS Out: Pulse Per Second Out – indicates the card is sending a clock synchronization signal.
LED 8	PPS In: Pulse Per Second In – indicates the card is receiving an external clock synchronization signal.

Figure

Figure 4-2 shows the correct LED state for the DAG 3.8S card without optical input.

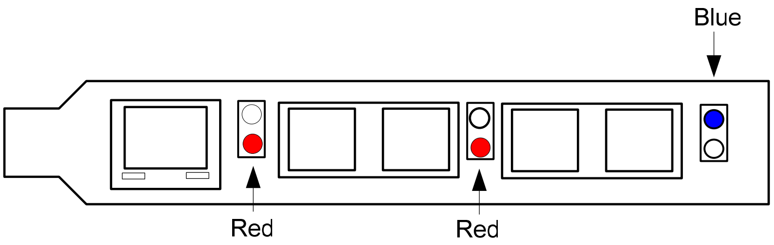

Figure 4-2. LED State for DAG 3.8S Card Without Optical Input.

4.2 DAG 3.8S Card Configuration

Description

The DAG 3.8S card uses two integrated SONET/SDH ATM/PoS physical layer interface devices to support capturing of ATM cells and HDLC encoded Packet-over-SONET data frames.

The card supports both OC3c, OC12c, STM-1 and STM-4c standards.

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

A successful DAG 3.8S card capture session is accomplished by receiver ports optical signal levels and checking the card is locked to stream data. This is followed by configuring the DAG card for normal use.

Procedure

Follow these steps for a successful DAG 3.8S card capture session.

Step 1. Check Receiver Ports Optical Signal Levels

The card supports 1300 nanometer singlemode and multimode fibre attachments with optical signal strength between 0 dBm and -22 dBm.

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

The card receiver ports are the lower of each dual-SC-style connectors, the closest to the PCI-X slot.

Cover card transmit ports with SC-style plugs to prevent dust and mechanical hazards damaging optics if not in use.

Step 2. Understand link layer configuration

Become knowledgeable of 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 seen at the host system.

Continued on next page

4.2 DAG 3.8S Card Configuration, continued

Procedure (continued)

Step 3. Check Card is Locked to Data Stream.

Configure card according to local settings.

Check through physical layer statistics that the card is locked to data stream.

The `dagthree` tool is supplied for both configuration and statistics of DAG 3.8S SONIC framers

Calling `dagthree` without arguments will list current settings. `dagthree -h` will print a help listing on the usage of the tool.

Before starting to configure the card, ensure the most recent pair of FPGA images are loaded on the card.

`dagthree -h` prints a help listing on tool usage.

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

Step 4. Load Latest Available PCI-X FPGA Image

```
dag@endace:~$ dagrom -rvp –d dag0 -f xilinx/dag38pcix-erf.bit
```

Step 5. Load Latest Available PHY FPGA Image

```
dag@endace:~$ dagld –x –d dag0 –f xilinx/dag38pp-erf.bit
```

Step 6. Display Card Configuration

Running the `dagthree` tool without arguments displays the card configuration.

```
dag@endace:~$ dagthree -d dag0
linkA   PoS noreset OC3c lt1 nodcr nolt0 fcl noeql enablea
linkB   PoS noreset OC3c lt1 nodcr nolt0 fcl noeql enableb
sonetA  scramble slave
sonetB  scramble slave
posA    nopmax nopmin nocrc pscramble
posB    nopmax nopmin nocrc pscramble
packet  varlen slen=48 noalign64
packetA drop=0
packetB drop=0
pcix    66MHz 64-bit buf=32MB rxstreams=1 txstreams=1 mem=0:0
```

4.3 Configuration in WYSYCC Style

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

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

Process Follow these steps configure the DAG 3.8S card in what you see can change style.

Step 1. Configure to ATM

Type:

```
dag@endace:~$ dagthree -d dag0 ATM
linkA   ATM noreset OC3c lt1 nodcr nolt0 fcl noeql enablea
linkB   ATM noreset OC3c lt1 nodcr nolt0 fcl noeql enableb
sonetA  scramble slave
sonetB  scramble slave
atmA    ascramble noaidle noahec noapass noafix
atmB    ascramble noaidle noahec noapass noafix
packetA drop=0
packetB drop=0
pcix    66MHz 64-bit buf=32MB rxstreams=1 txstreams=1 mem=32:0
```

Step 2. Other Options

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

```
dag@endace:~$ dagthree -d dag0 nofcl
linkA   ATM noreset OC3c lt1 nodcr nolt0 nofcl noeql enablea
linkB   ATM noreset OC3c lt1 nodcr nolt0 nofcl noeql enableb
sonetA  scramble slave
sonetB  scramble slave
atmA    ascramble noaidle noahec noapass noafix
atmB    ascramble noaidle noahec noapass noafix
packetA drop=0
packetB drop=0
pcix    66MHz 64-bit buf=32MB rxstreams=1 txstreams=1 mem=32:0
```

4.4 DAG 3.8S Card Configuration Options

Description There are many DAG 3.8S card configuration options supported.

Description

`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 OC12c 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]acramble`	[un]set SONET scrambling
`master`	Generate SONET tx clock internally
`slave`	Drive SONET tx 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.
`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.
`[en|dis]ablea`	Enable or Disable Port A for capture
`[en|dis]ableb`	Enable or Disable Port B for capture
`[no]align64`	Generate records with 64-bit alignment [default 32-bit]
`mem=X:Y`	configure memory allocated to streams 0, 1,…..
`rxonly`	Assign all buffer memory to receive streams.
`txonly`	Assign all buffer memory to transmit streams.
`rxtx`	Assign buffer memory to transmit and receive streams.

4.4 DAG 3.8S 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.

Continued on next page

4.4 DAG 3.8S Card Configuration Options, continued

Status bits display (continued)

`lof` Loss of frame.

If set, `oof` had been asserted for more than 3 milliseconds.

`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
A:     0     0     0     0    0    0    0    0      cf
B:     0     0     0     0    0    0    0    0      cf
A:     0     0     0     0    0    0    0    0      cf
B:     0     0     0     0    0    0    0    0      cf
```

Continued on next page

4.4 DAG 3.8S Card Configuration Options, continued

ATM cell stream example

An example for an ATM cell stream at OC 12c is:

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

Port statistics

Statistics for each port are printed on alternate lines each second. To view statistics for one port only, use the –a or –b flags to dagthree.

Port statistics example

An example Port A statistics for an ATM cell stream at OC12c is:

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

The following example indicates a problem with optical light levels on both ports.

```
     los  bip3  bip2  bip1   lop  oof  lof  los  label    lcd   Sync
 A:    1     0     0     0     0    1    1    1     6d      1      0
 B:    1     0     0     0     0    1    1    1     6d      1      0
 A:    1     0     0     0     0    1    1    1     6d      1      0
 B:    1     0     0     0     0    1    1    1     6d      1      0
```

Card set to OC3 PoS

An example of a card set to OC3 PoS while the line carries OC12 PoS is:

```
     los  bip3   bip2   bip1   lop   oof   lof   los    label
A:     0     0      0      0     1     1     1     0       3b
B:     0     0      0      0     1     1     1     0       8f
A:     0     0      0      0     1     1     1     0       57
B:     0     0      0      0     1     1     1     0       F1
```

Continued on next page

4.4 DAG 3.8S Card Configuration Options, continued

Card set to OC12 PoS

An example of a card set to OC12 PoS while the line carries OC3 PoS is:

```
      los  bip3   bip2  bip1  lop   oof   lof   los    label
 A:    0     0      0     0     1     1    1     0       fe
 B:    0     0      0     0     1     1    1     0       8f
 A:    0     0      0     0     1     1    1     0       f1
 B:    0     0      0     0     1     1    1     0       Bf
```

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!

4.5 Verify DAG 3.8S Card Configuration

Description

The card configuration is verified as being correct by checking settings and 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.

Continued on next page

4.5 Verify DAG 3.8S Card Configuration, continued

Procedure (continued)

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_crc
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.

Continued on next page

4.7 Inspect Links Data and Cells, continued

Description, continued

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.

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.

```
dag@endace:~$ dagthree -d /dagN pos
dag@endace:~$ dagthree -d dag0 -c 11,12 -i
   pos_good_frames_A    pos_bytes_rcvd_A   pos_good_frames_B    pos_bytes_rcvd_B
             2683706            16777215             2683707            16777215
               72675             9674302               72764             9674178
               73226             9747316               73226              974322
               73225              974250               73225              974240
               73225              974630               73225              974640
```

4.8 Reporting Problems

Description

Support is provided with a service contract. If 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 `dagrom –x`.
9.	Physical layer status reported by: `dagthree`
10.	Network link statistics reported by: `dagthree –si`
11.	Network link configuration from the router where available.
12.	Contents of any scripts in use.
13.	Complete output of session where error occurred including any error messages from DAG tools. The `typescript` Unix utility may be useful for recording this information.
14.	A small section of captured packet trace illustrating the problem.

USE THIS SPACE FOR NOTES

5.0 RUNNING DATA CAPTURE SOFTWARE

Introduction

For a typical measurement session, the `scripts/dag38start` script is edited and used to operate the cards directly.

In this chapter

This chapter covers the following sections of information.

- Starting DAG 3.8S Card Capture Session
- High Load Performance
- DAG 3.8S Card Packet Transmission Capabilities

5.1 Starting DAG 3.8S Card 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 DAG 3.8S card’s physical layer is then set and checked.

Process

Starting a data capture session is described in the following process.

Process	Description
Slen parameter default setting.	Parameters are set with `dagthree`. Slen parameter is set by default to 48. If only part of a packet is required, such as for IP header capture, the value of `slen` can be changed using `dagthree`. `tools/dagthree slen=128 varlen`
Setting capture session parameters.	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 16 to 2040 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. For example, full packet capture: `tools/dagthree –d dag0 varlen slen=2040`

Continued on next page

5.1 Starting DAG 3.8S Card Capture Session, continued

Process,continued

Process	Description
Setting fixed length mode.	In fixed length mode, packets longer than the selected slen are truncated to slen. Packets shorter than slen produce records are padded out to the 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/dagthree –d dag0 novarlen slen=48`
Disabling individual ports.	The A and B ports can be individually enabled and disabled for capture using `dagthree`. `tools/dagthree –d dag0 disableb`
Starting capture session.	Once the capture parameters are configured, a capture session is started by: `tools/dagsnap –v –o tracefile` Option `-v` provides user information during capture; it can be omitted for automated trace runs. If the `–o 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. `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` flag.

5.2 High Load Performance

Description

As the DAG 3.8S card captures packets from the network link, it writes a record for each packet into a large buffer in the host PC's main memory.

Avoiding packet loss

In order to avoid packet loss, the user application reading the record, such as `dagsnap`, must be able to read records out of the buffer faster than they arrive, otherwise the buffer eventually fills, and packet records are lost.

For Linux and FreeBSD, when the PC buffer becomes full the message:

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

is displayed on the PC screen, and printed to `/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 can be detected in-band by observing the Loss Counter lctr field of the Extensible Record Format [ERF]. The Endace ERF is explained in Chapter 7 of this document.

Increasing buffer size

The host PC buffer can be increased to deal with bursts of high traffic load on the network link.

By default the dagmem driver reserves 32MB of memory per DAG card in the system. Capture at OC-12/STM-4 (622Mbps) rates and above may require a larger buffer.

128MB or more is suggested for Linux/FreeBSD.

For the DAG 3.8S card Windows operating system the upper limit is 128MB.

In Debian Linux the amount of memory reserved is changed by editing the file `/etc/modules`.

```
# For DAG 3.x, default 32MB/card
dagmem
#
# For DAG 4.x or 6.x, use more memory per card, E.G.
# dagmem dsize=128m
```

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

The value of `dsize` multiplied by the number of DAG cards must be less than the amount of physical memory installed, and less than 890MB.

5.3 DAG 3.8S Card Packet Transmission Capabilities

Description

The firmware included with the DAG 3.8S card allows the card to transmit as well as receive packets, however the DAG card does not appear as a network interface to the operating system.

Process

The following information describes the DAG capabilities of the DAG firmware for the transmission and receiving of packets.

Process	Description
Explicit packet transmission.	The DAG will not respond to ARP, ping, or router discovery protocols. It will only transmit packets explicitly provided by the user. This capability allows the DAG card to be used as a simple traffic load generator. The DAG can also be used to retransmit previously recorded packet traces. The packet trace will be transmitted at 100% line rate, the packet timing of the original trace file is not reproduced.
Dagflood utility	The `dagflood` utility can transmit ERF format packet traces. The ERF trace file to be transmitted must contain only ERF records of the type matching the current link configuration. The ERF records to be transmitted must all have a length which is a multiple of 64-bits. When capturing a packet trace for later transmission, you can set 64-bit alignment using the `dagthree align64` command.

Continued on next page

5.3 DAG 3.8S Card Packet Transmission Capabilities, continued

Process,continued

Process	Description

Convert trace files.	It is also possible to convert trace files that have been captured without the `align64` option. This can be done with the command:
	`dagconvert -v -i in.erf -o out.erf -A8`
	If uncertain that a trace file is 64-bit aligned for transmission with `dagflood`, the file can be tested with dagbits:
	`dagbits -vvc align64 -f tracefile.erf`
	If a captured trace file is not available, the `daggen` program is capable of generating trace files containing simple traffic patterns. This allows the DAG card to be used as a test traffic generator.

Capture received traffic while transmitting.	It is possible to capture received traffic while transmitting. Capture programs such as `dagsnap`, `dagconvert`, and `dagbits` can be used while `dagflood` is sending packets.
Configuring DAG card for transmission.	To configure a DAG card for transmission, some memory must be allocated to a transmit stream. In the `dagthree` output, `buf=nMB` indicates that `n` megabytes of memory has been allocated to this DAG card in total. This memory can be split between the available receive and transmit stream buffers. The memory allocation is displayed with `mem=X:Y`, where X is the amount of memory allocated to receive stream 0 in MB, and Y is the amount of memory allocated to transmit stream 1 in MB. By default the memory is evenly split between the receive streams, the transmit streams have no memory allocated. If the card is to be used only for transmit, the `dagthree txonly` option can be used to recover the receive buffer memory and assign all the memory to transmit.

Continued on next page

5.3 DAG 3.8S Card Packet Transmission Capabilities, continued

Process,continued

Process	Description
Configuring DAG card for transmission. [Continued]	If the card is to be used for both transmitting and receiving, the `rxtx` option can be used. This allocates 16MB of memory to each transmit stream, and divides the remaining memory between the receive streams. Alternatively the memory allocation can be directly set with `mem= X:Y` option. The stream buffer memory allocation can only be changed when no packet capture or transmission programs are running.

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 synchronization 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 synchronization, 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 synchronization 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.

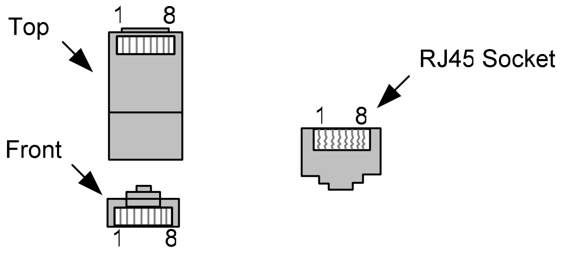

Figure 6-1. RJ45 Plug and Socket Connector Pin-outs.

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.

USE THIS SPACE FOR NOTES

7.0 DATA FORMATS OVERVIEW

In this chapter

This chapter covers the following sections of information.

- Data Formats
- Time Stamps

7.1 Data Formats

Description

The DAG 3.8S 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 variable length record.

timestamp		
timestamp		
type:1	flags	rlen
lctr		wlen
HDLC Header		
(rlen - 20) bytes of packet		

Table 7-1. Type 1 PoS HDLC 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 (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 [inrenal 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.

Table

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

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

Table 7-2. Type 2 Ethernet Variable Length Record.

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

7.2 Timestamps

Description

The ERF format incorporates a hardware generated timestamp of the packet's arrival.

The format of this timestamp is a single little-endian 64-bit fixed point number, representing seconds since midnight on the first of January 1970.

The high 32-bits contain the integer number of seconds, while the lower 32-bits contain the binary fraction of the second. This allows an ultimate resolution of 2^{-32} seconds, or approximately 233 picoseconds.

Another advantage of the ERF timestamp format is that a difference between two timestamps can be found with a single 64-bit subtraction. It is not necessary to check for overflows between the two halves of the structure as is needed when comparing Unix time structures, which are also available to Windows users from 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 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;
}
```