endace

accelerated

DAG 7.1S Card User Guide

EDM01-17

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

Overview

Endace DAG 7.1S card provides the means to transfer data at the full speed of the network into the memory of the host PC, with zero packet loss guaranteed in even worst-case conditions. Further, unlike a NIC, Endace products actively manage the movement of network data into memory without consuming any of the host PC's resources. The full attention of the CPU remains focused on the analysis of incoming data without a constant stream of interruptions as new packets arrive from the network. For a busy network link, this feature has a turbo-charging effect similar to that of adding a second CPU to the system.

The DAG 7.1S is a Network Monitoring Interface card specifically designed to provide high efficiency monitoring and transmission of ATM or POS traffic with precision timestamping capability on 4 x STM-1 or 2 x STM-2 interfaces

Purpose of this User Guide

Description

The purpose of this User Guide is to provide you with an understanding of the DAG card architecture and functionality and to guide you through the following:

- Installing the card and associated software and firmware,
- Configuring the card for your specific network requirements,
- Running a data capture session,
- Synchronising clock time,
- Data formats

You can also find additional information relating to functions and features of the DAG 7.1S card in the following documents which are available from the Support section of the Endace website at www.endace.com:

- EDM04-08 Configuration and Status API Programming Guide
- EDM04-13 SAR API Programming Guide
- EDM04-11 IXP Filter API Programming Guide
- EDM04-08 DAG IXP Filter Loader Programming Guide

This User Guide and the Linux and Window Guides are also available in PDF format on the Installation CD shipped with your DAG 7.1S card.

System Requirements

General

The minimum system requirements for the DAG 7.1S card are :

- PC, at least Intel Xeon 1.8GHz or faster
- Minimum of 256 MB RAM
- At least one free PCI-Express slot supporting at least one lane
- Software distribution requires 30MB free space
- 6GB for installation of Endace software, which is optional

Operating System

This User Guide assumes you are installing the DAG card in a PC which already has an operating system installed.

However for convenience, a copy of Debian Linux 3.1 (Sarge) is provided as a bootable ISO image on the CDs that is shipped with the DAG card.

To install either the Linux/FreeBSD or Windows operating system please refer to the following documents which are also included on the CD that is shipped with the DAG card.

- EDM04-01 Linux FreeBSD Software Installation Guide
- EDM 04-02 Windows Software Installation Guide

Other Systems

For advice on using an operating system that is substantially different from either of those specified above, please contact Endace Customer Support at support@endace.com

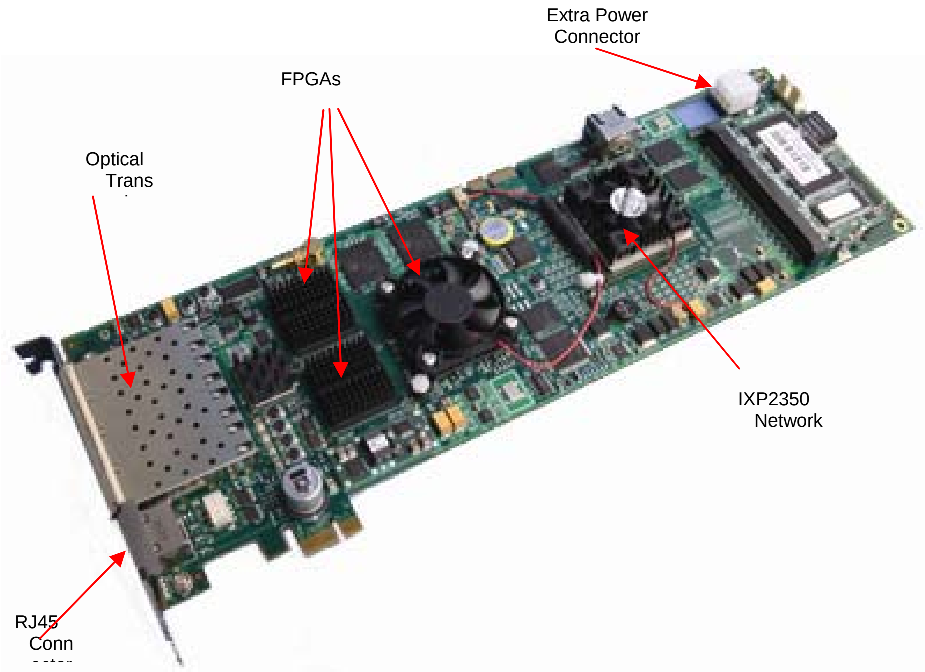
Card Description

The DAG 7.1S SDH/SONET Network Monitoring Card provides either four STM-1 (OC3) or two STM-2 (OC12) interfaces supporting concatenated or channelised ATM or Packet Over Sonet (POS) networks.

The DAG 7.1S has four optical transceivers which can be operated simultaneously.

The key features of the card are:

- Four interfaces allow full line rate capture and processing for 4 x STM-1/OC-3 or 2 x STM-2/OC-12.
- Fully programmable Intel IXP Network Processor
- PCI Express bus interface.
- 1244Mpps raw transmit and receive bandwidth.
- Combined FPGA and network processor architecture.
- Channelised and concatenated support.
- ATM AAL2 and AAL5 segmentation and reassembly.
- PoS IP filtering



Card Architecture

Serial SONET/SDH optical data is received by four optical interfaces, and passed through deserializers.

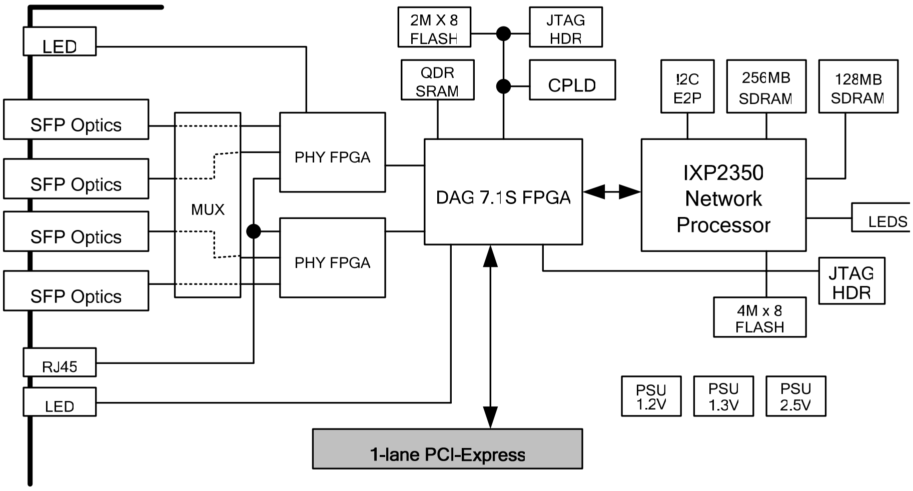
The network data feeds immediately into two physical layer FPGAs. The SONET/SDH payload data is then sent to the main FPGA.

The FPGA contains the packet record processor, PCI Express interface logic and the DAG Universal Clock Kit (DUCK) timestamp engine. The DUCK provides high resolution per packet timestamps which can be accurately synchronised. Time stamped packet records are then stored in the lower FIFO.

> **Note:** For further information on the DUCK and time synchronising please refer to Chapter 8: *Synchronising Clock Time* later in this User Guide.

An Intel IXP network processor is logically located next to the main FPGA. The main FPGA can route packets to either the IXP network processor for additional processing before routing onto the host or directly to the host via the PCI-Express port.

The following diagram shows the card's major components and the flow of data.



Extended Functions

In addition to standard packet capture the DAG 7.1S card also provides the following additional functionality:

TCP/IP Filtering and Classification
This feature allows you to classify packets into arbitrary categories with then drop, retransmit or capture a packet to the host based upon the result. You can also change filter rules “on the fly” with any loss of data

ATM Segmentation and Reassembly
This feature allows you to eliminate the significant CPU load associated with AAL2/AAL5 and reassembly on a busy ATM link by offloading this process to the DAG card. It also provides the ability to reduce volume of captured data to only what is required by filtering on VPI/VCI pairs

TCP/IP Filtering and Classification

The specifications for the IP filtering/packet classification are:

- Packets are classified and filtered by IP header (both IPv4 and IPv6) and/or UDP/TCP/SCTP port number.
- Up to 1024 IP header classification rules.
- Up 254 UDP/TCP/SCTP port or ICMP type rules can set per IP header classification.
- Classification rules are assigned a user-defined 14-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.

AAL5/AAL3 Reassembly

The ATM AAL5 Reassembler specifications are:

- Supports up to 8192 simultaneously active VCI/VPI/CIDs
- Supports simultaneous reassembly of AAL2 and AAL5 frames up to 8KB long.
- VPI/VCI/CID scanning
- Supports up to full OC-12/STM-4 cell rate on two interfaces simultaneously [~2.8 million cells/sec], or four full OC-3/STM-1 interfaces for AAL5 reassembly.
- Supports up to 4 x OC-3/STM-1 cell rate on combined four interfaces [~3.5 thousand cells/sec] for AAL2 reassembly.
- Optional ATM cell filtering prior to reassembly.

Chapter 2: Installation

Introduction

A DAG 7.1S card can be installed in any free PCI-Express slot.

The DAG 7.1S card operates on a single lane PCI-Express, this interface is capable of providing a maximum throughput of 1.8GB/s for both receive and transmit.

You can run multiple DAG 7.1S cards on one bus. By default, the DAG driver supports up to four DAG cards in one system.

DAG Driver Device

The DAG device driver must be installed before you install the DAG card itself.

If you have not already completed this please follow the instructions in *EDM04-01 Linux FreeBSD Software Installation Guide* or E*DM 04-02 Windows Software Installation Guide* as appropriate, which are included on the CD shipped with the DAG card.

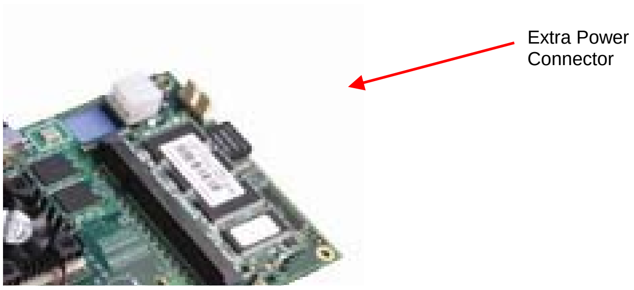
Inserting the DAG Card

To insert the DAG card in the PC follow the steps described below:

- Turn power to the computer OFF,
- Remove the PCI bus slot screw and cover,
- Insert DAG card into PCI-e bus slot ensuring that it is firmly seated in the slot,
- Check the free end of the card fits securely into the card-end bracket that supports the weight of the card,
- Secure the card with the bus slot screw,

- Connect the extra power connector located on the top edge of the card.

  Note: Ensure you do this before powering up the computer. Failure to do so may cause damage to the card.

- 

- Turn power to the computer ON.

Port Connectors

The DAG 7.1S re has 4 SFP socket connectors. Each connector consists of an optical fibre transmitter and receiver.

The upper connector of each pair is used for the transmit signal. These can be connected to daisy-chain systems if you have facility loopback (fcl) set on the card. You can also connect them if you are using a data generation programme.

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

There is an 8-pin RJ-45 socket located below the optical port connectors on the car bracket. This is available for connection to an external time synchronisation source.

! **Caution:** Never connect an Ethernet network or telephone line to the RJ-45 sockets.

Pluggable Optical Transceivers

Overview

The DAG7.1S card uses industry standard Small Form-factor Pluggable (SFP] optical transceivers.

The transceivers consists of two parts:

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

! **Note:** You must select the correct transceiver type to match the optical parameters of the network to which you want t connect. Configuring the card with the wrong transceiver type may damage the card.

You can connect the transceiver to the network via LC-style optical connectors.

For further information on Pluggable Optical Transceiver please refer to the Endace website at www.endace.com/dagpluggable.htm.

Setting Power

The optical power range depends on the particular SFP module that is fitted to the DAG card.

However Endace recommends the SFP modules described below which can be supplied with the DAG 7.1S card:

Manufacturer	Part number	Mode
Finisar	FTRJ1322	OC-12, Single Mode
Finisar	FTRJ1323	OC-3, Single Mode

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

The optical power is always a negative value, indicating power that is less than 1 mW. The most sensitive devices can work at power levels down as low as –30dBm or 1µW.

The DAG 7.1S card optical power module configuration for Multi Mode Fibre (MMF)) and Single Mode Fibre (SMF) is shown below:

Part #	Fibre	Data Rate	Max Pwr [dBm]	Min Pwr [dBm]	Nom Pwr [dBm]
FTRJ1322	SMF	622	-8	-28	-
FTRJ1323	SMF	155	-8	-28	-
-	MMF	622	-	-	-
-	MMF	155	-	-	-

Power Input

The optical power input to the DAG card must be within the receiver's dynamic range of 0 to -22dBm. If it is slightly outside of this range it will cause an increased bit error rate. If it is well outside of this range the system will not be able to lock onto the SONET signal.

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. In extreme cases, excess power can damage the receiver.

When you set up the DAG card measure the optical power at the receiver and ensure that it is well within the specified power range.

To adjust the input power:

- Change the splitter ratio if power is too high or too low, or
- Insert an optical attenuator if power is too high.

Splitter Losses

Splitters have the insertion losses either marked on their packaging or described in their accompanying documentation. General guidelines are:

- 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

Note: A single mode fibre connected to a multi-mode input has minimal extra loss. A multi-mode fibre connected to a single mode input creates large and unpredictable loss.

Chapter 3:
Configuring the Card

Introduction

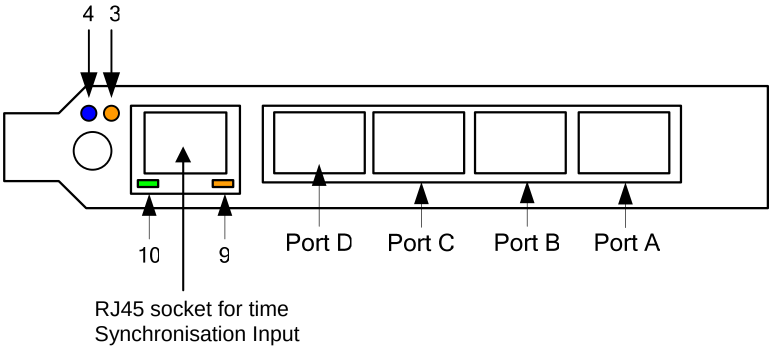
Configuring the DAG card for data capture involves the following steps:

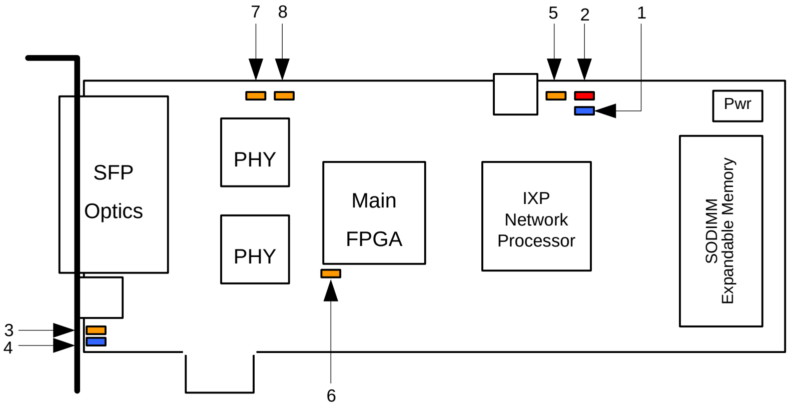
- Loading an image and programming the FPGA,
- Setting the link,
- Checking the link,
- Configuring the connections,
- Capturing data.

The `dagchan` tool which is also supplied with the DAG card allows you to configure channel characteristics. Sample `dagthree` and `dagchan` outputs are shown later in this chapter.

LEDs and Inputs

Before you begin to configure the DAG card it is important to understand the function of the various LEDs associated with the card, as well as the sockets on the PCI bracket.





The LED functions are described below

LED	Description
1	IXP Network Processor Status. When the IXP is booted this LED will

	flash at 1 sec intervals. The IXP does not automatically boot when the card is powered ON.
2	IXP status indication - depends on the extra functionality installed
3	PCI burst manager status – should be on after the card is configured.
4	PCI FPGA status – should be on soon after the computer is started
5	IXP status indication - depends on the extra functionality installed.
6	Unused
7	PP Image loaded
8	PP Image loaded
9	PPS Out: Pulse Per Second Out – indicates the card is sending a clock synchronization signal
10	PPS In: Pulse Per Second In – indicates the card is receiving an external clock synchronization signal.

Concatenated Configuration

Overview

The DAG 7.1S card uses four integrated SONET/SDH ATM/PoS physical layer interface devices to support capturing of ATM cells or HDLC encoded Packet-over-SONET data frames.

The card supports unchannelized POS/ATM; OC-3c, OC-12c, STM-1c and STM-4c standards for transmit and receive. Additionally channelized ATM/HDLC receive is supported on a single port only.

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 card capture session is accomplished by checking the 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. These steps are described next.

Check Receiver Ports Optical Signal Levels

The card supports 1300 nanometreer 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-LC-style connectors, the closest to the PCI-Express slot.

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

Understand Link Layer Configuration

Become knowledgeable of the link layer configuration in use at the network link being monitored. Important parameters include OC-3 vs. OC-12 configuration, Channelized vs. Concatenated, 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.

Load Latest Available PCI-Express FPGA Image

```
dag@endace:~$ dagrom -rvp –d dag0 -f xilinx/ dag71spci-conc-terf.bit
```

Load Latest Available PHY FPGA Images

```
dag@endace:~$ dagld –x –d dag0
        xilinx/dag71pp-erf.bit:xilinx/ dag71spp-conc-terf.bit
```

Concatenated Configuration (cont.)

The table below shows the available configurations.

> **Note:** Not all configurations are available within a single FPGA image.

Image	Number of Ports	Port Type	VC Type and Number	Demapper
1	4	STM-1c	4 x VC-4 4 x VC-4-4c	PoS ATM
2	1	STM-1	252 x VC-12 (E1) 336 x VC-11 (T1)	ATM Bit-HDLC RAW
	1	STM-4	504 x VC-12 (E1) 672 x VC-11 (T1)	ATM Bit-HDLC RAW

The DAG 7.1S card has four optical transceivers which can be operated simultaneously.

The integrated embedded processor can be used for PoS packet filtering or AAL2/AAL5 segmentation and reassembly.

Display Card Configuration

Running the dagconfig tool without arguments displays the card configuration.

```
dagconfig -d dag0
SFP A: laser detect nosignal nosfppwr
SFP B: laser detect nosignal nosfppwr
SFP C: laser detect nosignal nosfppwr
SFP D: laser detect nosignal nosfppwr

Port status
Port A: nolock oc12 core_on nofifo_error slave
Port B: nolock oc12 core_on nofifo_error slave
Port C: nolock oc12 core_on nofifo_error slave
Port D: nolock oc12 core_on nofifo_error slave

SONET/SDH status
SONET A: oc3 vc3 scramble tu11 async
SONET B: oc3 vc3 scramble tu11 async
SONET C: oc3 vc3 scramble tu11 async
SONET D: oc3 vc3 scramble tu11 async

E1/T1 status
E1/T1 A: no_payload notxais
E1/T1 B: no_payload notxais
E1/T1 C: no_payload notxais
E1/T1 D: no_payload notxais
```

Concatenated Configuration (cont.)

Display Card Configuration (cont.)

```
Phy status (AMCC1213):
eql fcl noreset

Concatenated Demapper Status:
pscramble crc32 atm noaidle
pscramble crc32 atm noaidle
pscramble crc32 atm noaidle
pscramble crc32 atm noaidle
Concatenated Mapper Status:
pscramble crc32 atm
pscramble crc32 atm
pscramble crc32 atm
pscramble crc32 atm

GPP:
varlen slen=48 align64
PCI Burst Manager
33Mhz buffer size = 32 rx_streams = 1 tx_streams = 1 mem=0:0
```

Verify Optical Signal is Received

Those ports which receive a valid optical signal are indicated on dagconfig output. To have a valid optical signal, both ends of the link must use the same type of optical transceivers. Also the optical fiber used must match the optical transceiver requirements and must be in good condition.

SFP A: laser detect signal nosfppwr

SFP B: laser detect nosignal nosfppwr

SFP C: laser detect nosignal nosfppwr

SFP D: laser detect signal nosfppwr

Verify Mapping/Framing is Setup Correctly

Both ends of the link need to match the framing and mapping parameters. When the frequency is setup correctly, the `lock` keyword is enabled. The main parameters involved to set the framing are:

- The link speed: `oc3 / oc12`
- The clock master: `master / slave`
- The payload mapping itself: `vc4c / vc4 / vc3`

Sometimes the framer/mapper requires a reset in order to synchronize after a configuration change. In that case the argument `reset` should be used in `dagconfig`.

When a channelized framing is chosen, a channelized firmware is needed. When a concatenated (or clear) framing is chosen, a concatenated firmware is needed.

Note This card allows different ports to have different configurations.

```
Port status
Port A: lock oc12 core_on nofifo_error slave
Port B: nolock oc12 core_on nofifo_error slave
Port C: nolock oc12 core_on nofifo_error slave
Port D: lock oc12 core_on nofifo_error slave
SONET/SDH status
SONET A: oc12 vc4c scramble tu11 async
SONET B: oc12 vc4c scramble tu11 async
SONET C: oc12 vc4c scramble tu11 async
SONET D: oc12 vc4c scramble tu11 async
```

Configuration Options

There are many DAG 7.1S card configuration options supported.

`atm`	Set concatenating mapper/demapper into ATM cell receive mode
`pos`	Set concatenating mapper/demapper into Packet-over-SONET (PoS) mode
`reset`	Reset the link
`oc3`	Set framer to OC-3 receive mode
`oc12`	Set framer to OC-12 receive mode
`vc4`	Set payload mapping to vc4
`vc4c`	Set payload mapping to vc4-4c
`vc3`	Set payload mapping to vc3
`tu11`	Ternary unit 11
`tu12`	Ternary unit 12
`[no]fcl`	[un]set facility loop back. This is useful for card chaining
`[no]eql`	[un]set equipment loop back. Do not touch
`[no]scramble`	[un]set SONET scrambling
`master`	Generate SONET tx clock internally
`slave`	Drive SONET tx clock from rx clock
`[no]pscramble`	[un]set Payload Scramble
`nocrc`	No PoS CRC checking
`crc16`	PoS CRC16 checks enabled
`crc32`	PoS CRC32 checks enabled
`[no]aidle`	When set pass through received idle cells
`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.
`sfppwr`	Turns on SFP module power for transmit.
`[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
`[en|dis]ablec`	Enable or Disable Port C 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.
`coreon`	TBD
`coreoff`	TBD
`e1_crc`	TBD
`e1_unframed`	TBD
`t1`	TBD
`t1_esf`	TBD
`t1_sf`	TBD
`t1_unframed`	TBD

Configuration in WYSYCC Style

Overview

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

Running the command '`dagconfig'` alone shows the current configuration. Each of the items displayed can be changed as shown below:

ATM Configuration

To configure the card to ATM mode use the following parameters. Replace `oc12` and `vc4c` to match the network configuration. The `lock` keyword under `‘Port status’` is set if the link frequency is correct.

It is important to note that the order of the command line options is considered, therefore `default` and/or `reset` options should be the first.

```
dagconfig –d dag0 reset default atm oc12 vc4c sfppwr slen=128

SFP A: nolaser detect nosignal nosfppwr
SFP B: nolaser detect nosignal nosfppwr
SFP C: nolaser detect nosignal nosfppwr
SFP D: nolaser detect signal sfppwr

Port status
Port A: nolock oc12 core_on nofifo_error slave
Port B: nolock oc12 core_on nofifo_error slave
Port C: nolock oc12 core_on nofifo_error slave
Port D: lock oc12 core_on nofifo_error slave

SONET/SDH status
SONET A: oc12 vc4c scramble tu11 async
SONET B: oc12 vc4c scramble tu11 async
SONET C: oc12 vc4c scramble tu11 async
SONET D: oc12 vc4c scramble tu11 async

E1/T1 status
E1/T1 A: no_payload notxais
E1/T1 B: no_payload notxais
E1/T1 C: no_payload notxais
E1/T1 D: no_payload notxais

Phy status (AMCC1213):
eql nofcl noreset

Concatenated Demapper Status:
pscramble crc32 atm noaidle
pscramble crc32 atm noaidle
pscramble crc32 atm noaidle
pscramble crc32 atm noaidle

Concatenated Mapper Status:
pscramble crc32 atm
pscramble crc32 atm
pscramble crc32 atm
pscramble crc32 atm

PP:
varlen slen=128 align64

PCI Burst Manager
33Mhz buffer size = 32 rx_streams = 1 tx_streams = 1 mem=16:16
```

PoS Configuration

To configure the card to PoS mode use the following parameters. Replace `oc12` and `vc4c` to match the network configuration. The `lock` keyword under ‘`Port status`’ is set if the link setup is correct.

```
dagconfig –d dag0 reset default pos oc12 vc4c sfppwr slen=128

SFP A: nolaser nodetect nosignal nosfppwr
SFP B: nolaser nodetect nosignal nosfppwr
SFP C: nolaser detect signal sfppwr
SFP D: nolaser detect signal sfppwr

Port status
Port A: nolock oc12 core_on nofifo_error slave
Port B: nolock oc12 core_on nofifo_error slave
Port C: lock oc12 core_on nofifo_error slave
Port D: lock oc12 core_on nofifo_error slave


SONET/SDH status
SONET A: oc12 vc4c scramble tu11 async
SONET B: oc12 vc4c scramble tu11 async
SONET C: oc12 vc4c scramble tu11 async
SONET D: oc12 vc4c scramble tu11 async

E1/T1 status
E1/T1 A: no_payload notxais
E1/T1 B: no_payload notxais
E1/T1 C: no_payload notxais
E1/T1 D: no_payload notxais

Phy status (AMCC1213):
eql nofcl noreset

Concatenated Demapper Status:
pscramble crc32 pos noaidle
pscramble crc32 pos noaidle
pscramble crc32 pos noaidle
pscramble crc32 pos noaidle

Concatenated Mapper Status:
pscramble crc32 pos
pscramble crc32 pos
pscramble crc32 pos
pscramble crc32 pos

GPP:
varlen slen=48 align64

PCI Burst Manager
33Mhz buffer size = 256 rx_streams = 1 tx_streams = 1 mem=240:16
```

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:~$ dagconfig -d dag0 –s1
```

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 `-s` 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.

`b3,b2, b1` 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 b's, the OCx carrier configuration is incorrect. Otherwise it indicates a signal problem related to either low light levels reaching the optical receivers, 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 OC-3 vs. OC-12 setting.

`oof` Out of frame.

If set, the section overhead processor is not locked to the SONET stream. It may indicate incorrect OC-3 vs. OC-12 setting.

`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.

`c2` 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.

`rei` Remote error indicator.

PoS OC12 Stream Example

An example for a card locked to a PoS OC-12 stream is on ports C and D, ports A and B are unused:

```
Port   LOS   LOF   OOF   B1   B2   B3   REI    C2      PTR
 A:     1     1     1     0    0    0    0   0xe800    lop
 B:     1     1     1     0    0    0    0   0xe800    lop
 C:     0     0     0     0    0    0    0   0xcf00   valid
 D:     0     0     0     0    0    0    0   0xcf00   valid
```

An example of a card set to OC-3 PoS while the line carries OC-12 PoS is:

```
Port   LOS   LOF   OOF   B1   B2   B3   REI    C2      PTR
 C:     1     0     1     1    1    1    1   0xfd00    lop
 D:     1     0     1     1    1    1    1   0x0100    lop

Port   LOS   LOF   OOF   B1   B2   B3   REI    C2      PTR
 C:     1     0     1     1    1    1    1   0xe200    Lop
 D:     1     0     1     1    1    1    1   0x5300    Lop
```

ATM Cell Stream Example

An example for an ATM cell stream at OC-12 is, ports A and B are unused:

```
Port   LOS   LOF   OOF   B1   B2   B3   REI    C2      PTR
 A:     1     1     1     0    0    0    0    0xe800    lop
 B:     1     1     1     0    0    0    0    0xe800    lop
 C:     0     0     0     0    0    0    0    0x1300   valid
 D:     0     0     0     0    0    0    0    0x1300   valid
```

No error bits are raised in dagconfig –s1 if the card is configured to PoS when on an ATM link or vice versa.

Network is ATM :

- C2 label should be `13`
- PTR should be `valid`
- Rest of the statistics should be `0`

Network is PoS

If network is PoS, the:

- C2 label should be 16 for PPP or cf for HDLC
- PTR should be `valid`
- Rest of the statistics should be `0`

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

Verify Configuration

The card configuration is verified as being correct by checking settings and path label for any errors as described in the following steps:

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

Ensure oof and lof are zero, otherwise change OC-3 settings to OC-12 or vice versa.

Ensure no bit interleaved parity errors occur, otherwise check cabling and light levels.

Ensure path label (C2) is correct as per the payload

Ensure ATM lcd is off and sync set.

Ensure PoS scrambling and CRC settings are correct.

Chapter 4:
Capturing Data

Starting a Session

For a typical data capture session follow the steps listed below:

- Move to the `dag` directory,
- Load the appropriate driver,
- Then load the appropriate FPGA image to each DAG card. For example, for ATM or PoS capture with one DAG 7.1S card installed use:

```
dagrom -rvp –d dag0 -f xilinx?????????????
```

- Set the integrity of the card's physical layer and check the integrity of the physical layer to each DAG card. For example:

```
dagthree –d0 dag0 default
```

- Start the capture session using:

```
tools/dagsnap -d dag0 –v -o tracefile
```

> **Note:** You can use the `-v` option to provide user information during a capture session although you may want to omit it for automated trace runs.

By default `dagsnap` will run indefinitely but can be stopped using *CTRL+C*. You can also configure `dagsnap` to run for a fixed time period then exit.

Setting Captured Packet Length

Snaplength

You can use `dagconfig` to set the length of the packets you want to capture.

By default the snaplength (`slen`) which is the portion of the packet that you want to capture, is set to 48. This means that only the first 48 bytes of each packet will be captured. If for example you want to capture only the IP headers you may want to set the length to a smaller value. Alternatively if you want to ensure you capture the whole packet you may want to set the length to a larger value.

> **Note:** The snaplength value must be a multiple of 4 and in the range of 16 to 2040.

Variable/Fixed Length

The DAG card is able to capture packets in two ways. They are:

- Variable length capture (varlen),
- Fixed length capture (novarlen).

In variable length (varlen) mode the whole packet will be captured providing its size is less than the slen value. Therefore to use this capture mode effectively you should set the slen value o the largest number of bytes that a captured packet is likely to contain.

Any packets that are longer than the slen value will be truncated to the length, while packets shorter than the slen value will produce shorter records which save bandwidth and storage space.

The example below shows configuration for variable length full packet capture:

```
dagconfig –d dag0 varlen slen=2040
```

In fixed length (novarlen) mode any packets that are longer than the slen value are truncated to that length, in the same way as for varlen capture. However any packets that are shorter than the slen value will produce records that are padded out to the slen length.

The example below shows configuration for fixed length 64-byte records, choose slen=48 (64 – ERF header size of 16) is:

```
dagconfig –d dag0 novarlen slen=48
```

Note: In novarlen mode you should avoid large slen values because short packets arriving will produce records with a large amount of padding which wastes bandwidth and storage space.

Enabling/ Disabling Ports

You can also use dagconfig to enable and disable individual ports for capture:

```
dagconfig –d dag0 disable
```

High Load Performance

Overview

As the DAG 7.1S card captures packets from the network link, it writes a record for each packet into a large buffer in the host PC's main memory.

Avoiding Packet Loss

To avoid packet loss, the user application reading the record, such as `dagsnap`, must be able to read records out of the buffer faster than they arrive. If not the buffer will eventually fill and packet records will be lost.

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.

In Linux and Free BSD, when the PC buffer fills, the following message displays on the PC screen:

```
        kernel: dagN: pbm safety net reached 0xNNNNNNNN
```

The same message is also printed to log /var/log/messages. In addition, when the PC buffer fills the "Data Capture" LED on the card will flash or flicker, or may go OFF completely.

In Windows no screen message displays to indicate when the buffer is full. Please contact Endace Customer Support at support@endace.com for further information on detecting buffer overflow and packet loss in Windows

Detecting Packet Losses

Once the buffer fills, any new packets arriving will be discarded by the DAG card until some data is read out of the buffer to create free space.

You can detect any such losses by observing the Loss Counter `(lctr` field) of the Extensible Record Format [ERF]. See *Chapter 6: Data Formats* later in this User Guide for more information on the Endace ERF.

Increasing Buffer Size

You can increase the size of the host PC buffer to enable it to cope 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. However if you are capturing at OC-12/STM-4 (622Mbps) rates or above, you may require a larger buffer.

For Linux/BSD 128MB or more is recommended. However you can change the amount of reserved memory by editing the file `/etc/modules` as follows

```
# 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
```

For Windows the upper limit is 32MB. This is usually sufficient however if you do need to increase the amount of reserved memory please contact Endace customer support at support@endace.com for more information

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

Note: The value of dsize multiplied by the number of DAG cards in the system must be less than the amount of physical memory installed, as well as less than 890MB.

Transmitting

The DAG 7.1S is able to transmit as well as receive packets and it is possible to . However the DAG card does not appear as a network interface to the operating system.

It is possible to capture received traffic while also transmitting. You can use capture tools such as `dagsnap`, `dagconvert`, and `dagbits` while `dagflood` is sending packets.

Configuring for Transmission.

To configure the DAG card for transmission, you must allocate some memory to a transmit stream.

In the `dagconfig` output, `buf=nMB` indicates that n mebibytes of memory has been allocated to the DAG card in total. You can split this allocation between the receive and transmit stream buffers which is displayed as a ratio as `mem=X:Y`, where X is the memory allocated in MB to the rx stream "0" and Y is the memory allocated to tx stream "1" in MB.

By default, memory is evenly split between the rx streams, with the transmit streams having no memory allocated.

If you wish to use the card for both transmitting and receiving, youshould use the `rxtx` option. This allocates 16MiB of memory to each transmit stream, and divides the remaining memory between the receive streams. Alternatively you can set the memory allocation directly using the mem= X:Y option.

> **Note:** You can only change the stream memory allocations when no packet capture or transmission is in progress.

Explicit Packet Transmission

The DAG card will not respond to ARP, ping, or router discovery protocols. It will only transmit packets explicitly provided by the user.

This capability allows you to use the DAG card as a simple traffic load generator. It can also be used to retransmit previously recorded packet traces.

The packet trace is transmitted at 100% line rate. The packet timing of the original trace file is not reproduced.

Dagflood Tool

The `dagflood` tool 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 length of the ERF records to be transmitted must be a multiple of 64-bits. When capturing a packet trace for later transmission, you can set 64-bit alignment using the `dagconfig align64` command.

Convert Trace Files

It is also possible to convert trace files that have been captured without the `align64` option. You can do this using:

```
dagconvert -v -i in.erf -o out.erf -A8
```

If you are unsure if a trace file is 64-bit aligned you can test the file using the `dagbits` tool

```
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.

Chapter 5: Synchronizing Clock Time

Overview

The Endace DAG cards have sophisticated time synchronisation capabilities, which allow for high quality timestamps, optionally synchronized to an external time standard.

The core of 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. The 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 designed to reduce time variance between sets of DAG cards or between DAG cards and coordinated universal time [UTC].

You can obtain an accurate time reference by connecting an external clock to the DAG card using the time synchronisation connector. Alternatively you can use the host PCs clock in software as a reference source without any additional hardware.

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

Common Synchronization

The DAG card time 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 also provides the TDS 2 Time Distribution Server modules and the TDS 6 units that enable you to connect multiple DAG cards to a single GPS or CDMA unit.

For more information please refer to the Endace website at http://www.endace.com/accessories.htm , or the *TDS 2/TDS 6 Units Installation Manual*.

Timestamps

ERF files contains a hardware generated timestamp of each packet’s arrival. The arrival time can be either the point at which the start of the packet arrives (head) or the point at which the end of the packet arrives (tail).

See *Default Configuration* in *Chapter 3: Card Configuration* earlier in this user guide for more information on configuring the timestamp head/tail option

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

The ERF timestamp allows you to find the difference between two timestamps using a single 64-bit subtraction. You do not need to check for overflows between the two halves of the structure as you would need to do when comparing Unix time structures.

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

Below is example code showing how a 64-bit ERF timestamp (erfts) can be converted into a struct timeval representation (tv):

```
unsigned long long lts;
struct timeval tv;

 lts = erfts;
 tv.tv_sec = lts >> 32;
 lts = ((lts & 0xffffffffULL) * 1000 * 1000);
 lts += (lts & 0x80000000ULL) << 1;      /* rounding */
 tv.tv_usec = lts >> 32;
 if(tv.tv_usec >= 1000000) {
     tv.tv_usec -= 1000000;
     tv.tv_sec += 1;
      }
```

Configuration Tools

The DUCK is very flexible, and can be used with or without an external time reference. It can accept synchronisation from several input sources, and also be made to drive its synchronisation output from one of several sources.

Synchronisation settings are controlled by the `dagclock` utility.

Note: You should only run `dagclock` after you have loaded the appropriate Xilinx images. If at any stage you reload the Xilinx images you must rerun `dagclock` to restore the configuration.

```
 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`	the DAG device
`-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

Note: By default, all DAG cards listen for synchronisation signals on their RS-422 port, and do not output any signal to that port

```
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
```

Card with Reference

Overview

To obtain the best timestamp accuracy you should connect the DAG card to an external clock reference, such as a GPS or CDMA time receiver.

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.

When the external time reference source is connected to the DAG card time synchronisation connector, the card automatically synchronises to a valid signal.

Pulse Signal from External Source

The DAG time synchronisation connector supports an RS-422 (PPS) signal from an external source. This is derived directly from an external reference source, or distributed through the Endace TDS 2 (Time Distribution Server) module which allows two DAG cards to use a single receiver. It is also possible for more than two cards to use a single receiver by "daisy-chaining" TDS-6 expansion modules to the TDS-2 module. Each TDS-6 , module provides outputs for an additional 6 DAG cards.

Synchronise to an external source as follows:

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

Connecting the Time Distribution Server

You can connect the TDS 2 module to the DAG card using standard RJ-45 Ethernet cable including existing RJ-45 building cabling. The TDS may be located up to 600m (2000ft) from the DAG card depending upon the quality of the cable used, possible interference sources and other environmental factors. Please refer to the *TDS2/TDS6 User Guide* for more in formation

! **Caution:** Never connect a DAG card and/or the TDS 2 module to active Ethernet equipment or telephone equipment.

Testing the Signal

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

To test the signal is being received correctly and has the correct polarity use the `dagpps` tool as follows:

```
dagpps –d dag0
```

`dagpps` measures the input state many times over several seconds, displaying the polarity and length of input pulse. The DAG 3.7T card also has an LED indicator for synchronisation (PPS) signals. See *Chapter 3: Configuring the Card* earlier in this User Guide for more information.

Single Card No Reference

When a single card is used with no external reference, the card can be synchronised to the host PC clock. Most PC clocks are not very accurate by themselves, but the DUCK drifts smoothly at the same rate as the PC clock.

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

The synchronisation achieved with this method is not as accurate as using an external reference source such as GPS.

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

```
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
```

Two Cards No Reference

Overview

If you are using two DAG cards in a single host PC with no reference clock, you must synchronise the cards using the same method if you wish to compare the timestamps between the two cards. You may wish to do this for example if the two cards monitor different directions of a single full-duplex link. You can synchronise the cards in two ways:

- One card can be a clock master for the second. This is useful if you want both cards to be accurately synchronised with each other, but not so for absolute time of packet time-stamps, or
- One card can synchronise to the host and also act as a master for the second card

Two Cards No Reference (cont.)

Synchronising with Each Other

Although the master card's clock will drift against UTC, the cards will still be locked together. This is achieved by connecting the time synchronisation connectors of both cards using a standard RJ-45 Ethernet cross-over cable.

Configure one of the cards as the master so that the other defaults to being a slave as follows:

```
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
```

Note: The slave card configuration is not shown as the default configuration will work.

Synchronising with Host

To prevent the DAG card clock time-stamps drifting against UTC, the master can be synchronised to the host PC's clock which in turn utilises NTP. This then provides a master signal to the slave card.

Configure one card to synchronize to the PC clock and output a RS-422 synchronization signal to the second card as follows:

```
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.

Connector Pin-outs

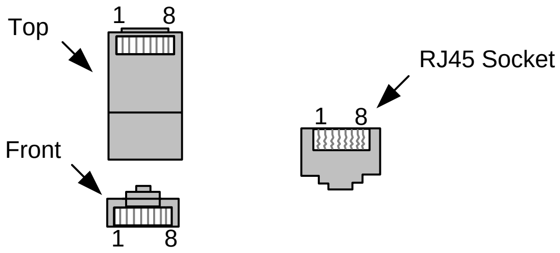
Overview

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

Pin Assignments

The 8-pin RJ45 connector pin assignments and plugs and sockets are shown below:

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



Normally you should connect the GPS input to the A channel input (pins 3 and 6).

The DAG card can also output a synchronization pulse for use when synchronizing two DAG cards without a GPS input. The synchronization pulse is output on the Out A channel (pins 1 and 2).

Ethernet Crossover Table

You can use a standard Ethernet crossover cable to connect the two cards as shown below:

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

Chapter 6: Data Formats

Overview

DAG Cards produce trace files in their own native format called ERF (Extensible Record Format). The ERF type depends upon the type of connection you are using to capture data.

The DAG 7.1S supports the following ERF Types:

ERF Type	Description
1	TYPE_HDLC_POS PoS with HDLC Framing
3	TYPE_ATM ATM Cell
4	TYPE_AAAL5 Reassembled AAL5 Frame
5	TYPE_MC_HDLC: Multi-channel HDLC Frame Record
7	TYPE_MC_ATM Multi-Channel ATM Cell Record
9	TYPE_MC-AAL5 Multi Channel AAL5 Frame Record
12	TYPE_MC-AAL2 Multi Channel AAL2 Frame Record
18	TYPE_AAL2 Reassembled AAL2 Frame

The ERF file contains a series of ERF records with each record describing one packet.An ERF file consists only of ERF records, there is no special file header which allows concatenation and splitting to be performed arbitrarily on ERF record boundaries.

Generic Header

All ERF records share some common fields. Timestamps are in little-endian (Pentium native) byte order. All other fields are in big-endian [network] byte order. All payload data is captured as a byte stream, no byte re-ordering is applied.

The generic ERF header for concatenated (clear) links is shown below.

<table>
<tr><th>Byte 3</th><th>Byte 2</th><th>Byte 1</th><th>Byte 0</th></tr>
<tr><td colspan="4">timestamp</td></tr>
<tr><td colspan="4">timestamp</td></tr>
<tr><td>type</td><td>flags</td><td colspan="2">rlen</td></tr>
<tr><td colspan="2">lctr</td><td colspan="2">wlen</td></tr>
<tr><td colspan="4">(rlen - 16) bytes of record</td></tr>
</table>

Generic Header (cont.)

The generic ERF header for channelised links is shown below:

Byte 3	Byte 2	Byte 1	Byte 0
timestamp			
timestamp			
type	flags	rlen	
lctr		wlen	
MCH (Multichannel header)			
(rlen - 20) bytes of packet			

timestamp The time of arrival of the cell, an ERF 64-bit timestamp. See *Timestamps* in *Chapter 6: Synchronising Clock Time* earlier in this User Guide for more information on timestamps.

type

One of the following:

1:	TYPE_HDLC_POS
3:	TYPE_ATM
4:	TYPE_AAAL5
5:	TYPE_MC_HDLC
7:	TYPE_MC_ATM
9:	TYPE_MC_AAL5
12:	TYPE_MC_AAL2
18:	TYPE_AAL2

flags

This byte is divided into several fields as follows:

1-0: Enumerates capture interface 0-3

2: Varying record lengths

3: Truncated record (insufficient buffer space)

4: RX error (link layer error)

5: DS error (internal error)

6: Reserved

7: General direction bit. This bit has two uses, it indicates from where a packet has arrived, either the host or line, and enables the XScale to target the packet at either the host or line. The direction bit can be interpreted in the context of either the Rx or Tx hole

In the XScale/Host Rx hole, a value of “1” indicates the ERF has arrived from the line. A value of “0” indicates the record was received from the host.

In the XScale Tx hole, a value of “1” tells the ERF Mux to direct packets to the line. Avalue of “0” directs packets to the host.

rlen

Record length. Total length of the record transferred over the PCI bus to storage.

lctr

Depending upon the ERF type this 16 bit field is either a loss counter of colour field. The loss counter records the number of packets lost between the DAG card and the memory hole due to overloading on the PCI bus.

wlen

Wire length. Packet length including some protocol overhead. The exact interpretation of this quantity depends on physical medium.

Type-5 Record

The format of the multichannel header for channelised links used for TYPE_MC_HDLC is shown below:

This header is divided into several bit fields as follows::

0-9: Connection number (0-511)

10-15: Reserved

16-23: Reserved

24: FCS Error

25: Short Record Error (<5 Bytes)

26: Long Record Error (>2047 Bytes)

27: Aborted Frame Error

28: Octet Error

29: Lost Byte Error

30: 1st Rec. This is the first record received since the connection was configured.

31: Reserved

Type-7 Record

The format of the multichannel header for channelized links used for TYPE_MC_ATM is shown below:

This header is divided into several bit fields as follows:

0-9: Connection number (0-1023) or IMA group ID

10-14: Reserved

15: Multiplexed from IMA into ATM stream

16-19: Physical port (0-15) cell was captured on

20-23: Reserved

24: Lost Byte Error

25: HEC corrected

26: OAM Cell CRC-10 Error (not implemented)

27: OAM Cell

28: 1st Rec. This is the first record received since the connection was configured.

29-31: Reserved

Type-9 Record

The format of the multichannel header for channelized links used for TYPE_MC_AAL5 is shown below:

This header is divided into several bit fields as follows:

0-9: Connection number (0-1023)

10-15: Reserved

16-19: Physical port (0-15) cell was captured on

20: CRC Checked

21: CRC Error

22: Length checked

23: Length Error

24-27: Reserved

28: 1st Rec. This is the first record received since the connection was configured.

29-31: Reserved

Type-12 Record

The format of the multichannel header for channelized links used for TYPE_MC_AAL2 is shown below:

This header is divided into several bit fields as follows:

0-9: Connection number (0-1023)

10-12: Reserved (possible extra connection numbers)

13-15: Reserved (indication of AAL2 type)

16-19: Physical port (0-15) cell was captured on

20: Reserved

21: 1st Rec. This is the first record received since the connection was configured.

22: MAAL Error

23: Length Error

24-31: Reserved

Chapter 7
Troubleshooting

Reporting Problems

If you have problems with a DAG card or Endace supplied software which you are unable to resolve, please contact Endace Customer Support at support@endace.com.

Supplying as much information as possible enables Endace Customer Support to be more effective in their response to you. The exact information available to you for troubleshooting and analysis may be limited by nature of the problem. However the following items will assist a quick resolution:

- DAG card[s] model and serial number.
- Host PC type and configuration.
- Host PC operating system version
- DAG software version package in use
- Any compiler errors or warnings when building DAG driver or tools
- For Linux and FreeBSD, messages generated when DAG device driver is loaded. These can be collected from command dmesg, or from log file /var/log/syslog.
- Output of daginf
- Firmware versions from dagrom –x.
- Physical layer status reported by: dagthree
- Network link statistics reported by: dagthree –si
- Network link configuration from the router where available.
- Contents of any scripts in use.
- Complete output of session where error occurred including any error messages from DAG tools. The `typescript` Unix utility may be useful for recording this information.
- A small section of captured packet trace illustrating the problem.