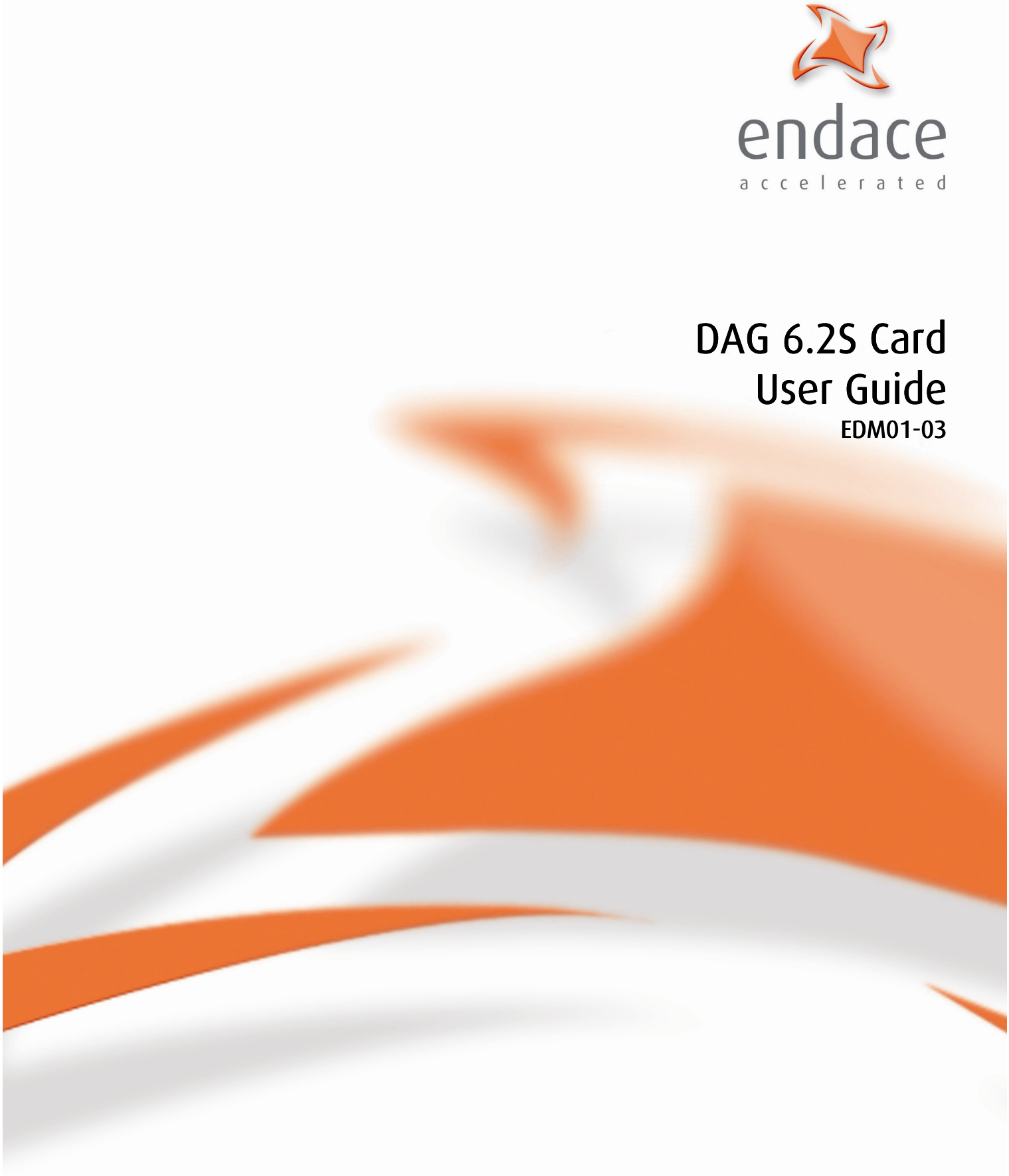

endace
accelerated

DAG 6.2S Card User Guide

EDM01-03

Published by:

Endace Measurement Systems® Ltd

Building 7
17 Lambie Drive

PO Box 76802
Manukau City 1702
New Zealand

Phone: +64 9 262 7260

Fax: +64 9 262 7261

support@endace.com

www.endace.com

International Locations

New Zealand
Endace Technology® Ltd

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

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

Americas
Endace USA® Ltd

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

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

Europe, Middle East & Africa
Endace Europe® Ltd

Sheraton House
Castle Park
Cambridge CB3 0AX
United Kingdom

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

Protection Against Harmful Interference

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

Extra Components and Materials

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

Disclaimer

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

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

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

Table of Contents

Chapter 1: Introduction

Introduction The installation of the Endace DAG 6.2S 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 6.2S Card User Manual is available on the installation CD.

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

- User Manual Purpose
- DAG 6.2S Card Product Description
- DAG 6.2S Card Architecture
- DAG 6.2S Card Extended Functions
- System Requirements

1.1 User Manual Purpose

Description The purpose of this DAG 6.2S Card User Manual is to describe:

-
- Setting Optical Power
- Confidence Testing
- Running Data Capture Software
- Synchronizing Clock Time
- Data Formats

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 6.2S Card Product Description

Description

The DAG 6.2S card performs packet capture from PoS OC-192c or STM-64c links, and 10 Gigabit Ethernet 10GBase-LR and 10GBase-LW links.

Figure 1-1 shows the DAG 6.2S PCI-X card.

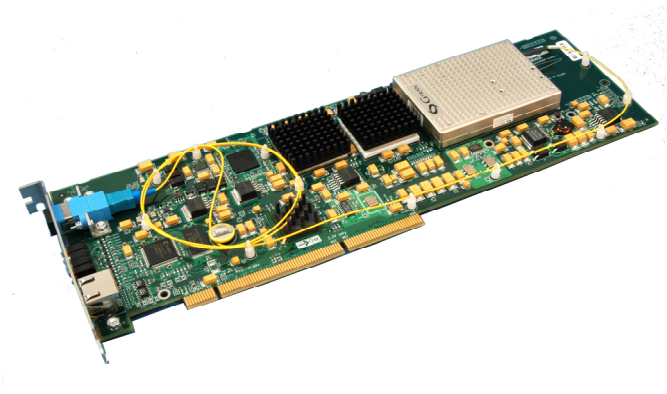

Figure 1-1. DAG 6.2S PCI-X Card.

The DAG 6.2S performs partial and full packet capture from PoS OC-192c or STM-64c links, and from 10 Gigabit Ethernet LAN 10GBase-LR and SONET/SDH WAN 10GBase-LW links.

1.3 DAG 6.2S Card Architecture

Description

Serial optical data is received by the optical interface, and fed into a physical layer ASIC.

The packet data is then fed immediately into the Rx FPGA. This FPGA contains the DUCK timestamp engine, and packet record processor.

Because of component close association, packets or cells are time-stamped accurately. Time stamped packet records are then stored in an external FIFO and passed into the PCI-X FPGA for transmission to the host.

Figure

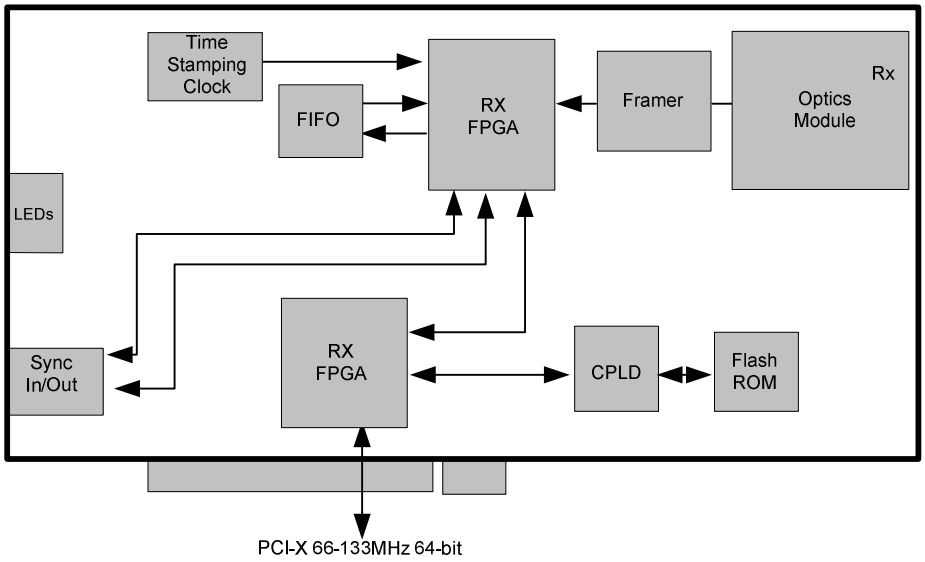

Figure 1-2. DAG 6.2S Card Major Components and Data Flow.

1.4 DAG 6.2S Card Extended Functions

Description

The shipped version of the DAG 6.2S does not contain a transmit path. It is intended to be used with fibre optic splitters.

Contact the Endace customer support team at support@endace.com to enable effective use of extended functions.

1.5 System Requirements

Description

The DAG 6.2S and associated data capture system minimum operating requirements are:

- PC, at least Intel Xeon 1.8GHz or faster
- Intel E7500, ServerWorks Grand Champion LE/HE or newer chip set
- 256 MB RAM
- At least one free PCI-X 1.0 slot supporting 66MHz operation
- Software distribution free space of 30MB

Operating system

Endace currently supports Windows XP, Windows Server 2000, Windows Server 2003, FreeBSD, RHEL 3.0,RHEL 4.0 and Debian Linux operating systems.

Different system

For advice on using a system substantially different from that specified above, contact Endace support at support@endace.com

Chapter 2: Installation

Introduction The DAG 6.2S card can be installed in any free PCI-X 1.0 slot. It operates in 66, 100, or 133MHz PCI-X mode, however it will not operate correctly in 32 or 64-bit PCI slots. Higher speed slots are recommended for best performance.

The DAG 6.2S should be the only device on the PCI-X bus if possible, as the cards make very heavy use of PCI-X bus data transfer resources.

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

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

- Installation of Operating System and Endace Software
- Insert DAG 6.2S Card into PC
- Card Port Connection

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 6.2S Card into PC

Description Inserting the DAG 6.2S card into a PC involves accessing the bus slot, fitting the card, and replacing the bus slot screw.

Procedure Follow these steps to insert the DAG 6.2S 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.

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

Step 3. Replace bus Slot Screw

Secure card with screw.

Step 4. Power Up Computer

2.3 Card Port Connection

Description

There is one square plastic SC-type optical connector for the received optical signal. There is no TX port on the DAG 6.2S card.

The DAG 6.2S card has an 8-pin RJ45 socket for time synchronization input. This socket should never be connected to an Ethernet network or telephone line.

Chapter 3: Setting Optical Power

Description

The optical power range depends on the particular device fitted on the DAG 6.2S card.

The DAG 6.2S card is shipped fitted with the GTRAN GT10-RXU 1310nm single mode receive module by default.

Optical power measure

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

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

Configuration

The following table describes the DAG 6.2S card optics power module configuration.

Part #	Fibre	Data Rate	Max Power [dBm]	Min Power [dBm]	Nominal Pwr [dBm]
GT10-RXU	SMF	10Gbps	0	-17	-9

In this chapter

This chapter covers the following sections of information.

- Optical Power Input
- Splitter Losses

3.1 Optical Power Input

Description

The optical power input to the DAG 6.2S 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.

Input power

When the DAG 6.2S card is set up, measure the optical power at the receiver and ensure 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

The GTIO-RXU transceiver uses 1310nm optics. Splitters used must be designed for 1310nm as the insertion loss will vary for different wavelengths. The not 1310nm type is never used.

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.

Chapter 4: Confidence Testing

Introduction The confidence testing is a process to determine the DAG 6.2S card is functioning correctly.

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

Interface statistics are also inspected during this process.

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

- Interpreting LED Status
- DAG 6.2S Card LED Display Functions
- Configuration in WYSYCC Style
- `dagsix` Utility
- Capture Session
- Inspect POS Interface Statistics
- Inspect 10G Ethernet Interface Statistics
- Reporting Problems

4.1 Interpreting LED Status

Description The DAG 6.2S card has 8 status LEDs, three blue, one green, one orange, one yellow, and two red. LEDs labelled RESERVED may be on or off under varying circumstances and these may be disregarded.

Figure Figure 4-1 shows the typical DAG 6.2S status LEDs.

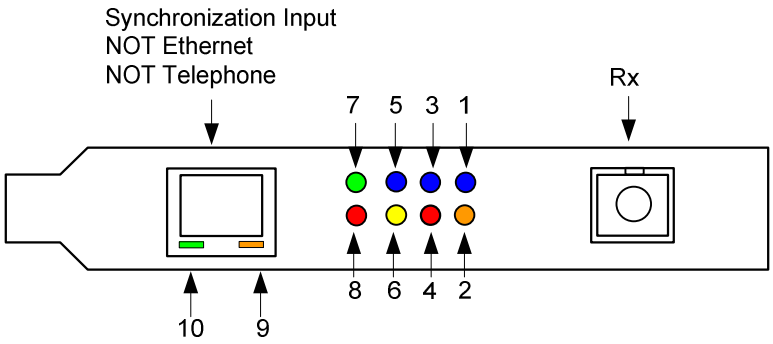

Figure 4-1. Typical DAG 6.2S Card Status LEDs.

LED on stages The following table describes the LED displays:

LED	Description
LED 1	RX FPGA successfully programmed.
LED 2	LOS: Loss of Signal – no valid optical signal seen by receiver.
LED 3	Reserved.
LED 4	Reserved.
LED 5	PCI-X FPGA successfully programmed
LED 6	Data capture in progress.
LED 7	Ethernet mode, OFF for PoS.
LED 8	Reserved.
LED 9	PPS Out: Pulse Per Second Out – indicates card is sending a clock synchronization signal
LED 10	PPS In: Pulse Per Second In – indicates card is receiving an external clock synchronization signal

4.2 DAG 6.2S Card LED Display Functions

Description When a DAG 6.2S card is powered up the blue coloured LED 5 should always come on, and the remaining LED's such as green and yellow display for specific functions.

Figure Figure 4-2 shows the correct LED state for DAG 6.2S card without optical input.

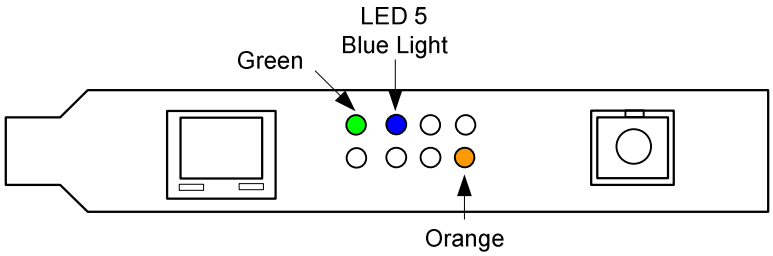

Figure 4-2. LED State for DAG 6.2S Card Without Optical Input.

4.3 Configuration in WYSYCC Style

Description

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

Running the command '`dagsix`' alone shows the current configuration. Each of the items displayed can be changed.

Configuration options

`default`	Set card to normal defaults.
`pos`	Set framer into Packet-over-SONET [PoS] mode.
`[no]lsfcl`	[un]set facility loopback on line side of phy.
`[no]lseql`	[un]set equipment loopback on line side of phy
`[no]fcl`	[un]set facility loopback on downstream side of phy.
`[no]eql`	[un]set equipment loopback on downstream side of phy.
`[no]pscramble`	[un]set Packet-over-SONET scrambling.
`[no]crc`	Dis/enable PoS CRC32 checking.
`[no]crcstrip`	[Do]Don’t include CRC in ERF record or wlen count.
`[no]pmin`	Ds/enable discard of packets smaller than a predefined minimum size.
`[no]pmax`	Dis/enable discard of packets larger than a predefined maximum size.
`long=X`	Maximum packet size for pmax.
`short=x`	Minimum packet size for pmin.
`slen=X`	Capture X bytes of packet data.
`eth`	Set framer to 10G Ethernet mode. Defaults to LAN.
`lan`	Set framer to Ethernet LAN mode 10GBase-LR
`wan`	Set framer to Ethernet WAN mode 10GBase-LW

Process Follow these steps configure the DAG 6.2S card in what you see is what you can change style.

Step 1. Check FPGA Images

Before configuring the DAG 6.2S card, ensure the most recent FPGA image is loaded on the card. Loading the newest available Rx FPGA image. This will cause LED 1 to light.

Load the newest available PCI-X FPGA image.

```
dag@endace:~$ dagrom -rvp -d dag0 -f xilinx/dag62pcix-erf.bit
```

Load the newest available Rx FPGA image. This will cause LED 1 to light.

```
dag@endace:~$ dagld -x -d dag0 xilinx/dag62rx-erf.bit

dag@endace:~$ dagsix -d dag0
link    noreset ETH nolsfcl nolseql nofcl noeql
Eth     lan crc crcstrip pmin pmax long=1518 short=64 nopscramble
packet  varlen slen=48 align64
packetA drop=0
pcix    133MHz 64-bit buf=128MiB rxstreams=1 txstreams=0 mem=0:0
Firmware: edag62sepcix_pos-erf_v2_9 2v1000fg456 2005/11/01 11:33:44
(user)
MAC Address: 00:00:00:00:00:00
```

Step 2. Configure DAG 6.2S Card for Normal Use

The `dagsix default` command is always used:

```
dag@endace:~$ dagsix -d dag0 default
 link    noreset POS nolsfcl nolseql nofcl noeql
PoS     crc crcstrip nopmin nopmax long=1502 short=9
packet  varlen slen=48 align64
packetA drop=0
pcix    133MHz 64-bit buf=128MiB rxstreams=1 txstreams=0 mem=128:0
Firmware: edag62sepcix_pos-erf_v2_9 2v1000fg456 2005/11/01 11:33:44
(user)
MAC Address: 00:00:00:00:00:00
```

NOTE: The default command also sets the card to PoS mode, which will cause LED 7 to go out. For Ethernet mode use `dagsix default eth lan` or `dagsix default eth wan` as appropriate.

NOTE: After loading the Rx FPGA firmware, the dagsix default command must be issued immediately to initialise the thermal management systems of the to prevent overheating and protective shutdown.

Step 3. Turn Pos Scrambling Off

Type:

```
dag@endace:~$ dagsix -d dag0 nopscramble
 link    noreset POS nolsfcl nolseql nofcl noeql
 PoS     crc crcstrip nopmin nopmax long=1502 short=9
 packet  varlen slen=48 align64
 packetA drop=0
 pcix    133MHz 64-bit buf=128MiB rxstreams=1 txstreams=0
 mem=128:0
Firmware: edag62sepcix_pos-erf_v2_9 2v1000fg456 2005/11/01
11:33:44  (user)
MAC Address: 00:00:00:00:00:00
```

Step 4. Set Configuration Options

Removing or adding the "no" prefix changes the configuration option settings.

Step 5. Select Configuration Option

Choose from complete list of configuration options supported:

`default`	Set card framer to normal defaults.
`pos`	Set framer into Packet-over-SONET [PoS] mode.
`[no]lsfcl`	[un]set facility loopback on line side of phy.
`[no]lseql`	[un]set equipment loopback on downstream side of phy.
`[no]fcl`	[un]set facility loopback on downstream side of phy.
`[no]eql`	[un] equipment loopback. This is for testing only.
`[no]pscramble`	[un]set Packet-over-SONET scrambling
`[no]crcstrip`	Do [not] include CRC in ERF record or wlen count.
`[no]crc`	Dis/enable PoS CRC32 checking.
`[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.
`long=X`	Maximum packet size for pmax.

Step 6. Select Configuration Option, continued

`short=x`	Minimum packet size for pmin.
`slen=X`	Capture X bytes of packet data.
`mem=X:Y`	Configure memory allocated to streams 0, 1, ..
`eth`	Set framer to 10G Ethernet made defaults to LAN.
`lan`	Set framer to Ethernet LAN made 10G Base-LR.
`wan`	Set framer to Ethernet WAN made 10G Base-LW

4.4 `dagsix` Utility

Description

The `dagsix` utility supports configuration status and physical layer interface statistics for DAG 6 series cards.

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

More details about the meaning of the various bits are supplied through the help page (`dagsix -h`) and below.

4.5 Capture Session

Description

A successful DAG 6.2S card capture session is accomplished by checking receiver port optical signal levels and checking the card has correctly detected the link. This is followed by configuring the DAG card for normal use.

Procedure

Follow these steps for a successful DAG 6.2S card capture session.

Step 1. Check Receiver Ports Optical Signal Levels.

The DAG 6.2S card supports 1310 nanometer singlemode fibre attachments with optical signal strength between 0 dBm and -17 dBm.

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

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

Step 2. Understand link layer configuration

Learn about the link layer configuration in use at the network link being monitored. Important parameters include specific scrambling options in use.

If the information cannot be obtained reliably, the card can be made to work by varying the parameters until data is arriving at the host system.

Step 3. Check Card is Locked to Data Stream.

Configure card according to local settings.

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

Step 4. List Current Settings.

The `dagsix` tool is supplied for the DAG 6.2S card framer configuration and statistics. Calling `dagsix` without arguments lists current settings. `The dagsix -h` prints a help listing on tool usage.

4.6 Inspect POS Interface Statistics

Description

Once the DAG 6.2S card has been configured for POS mode, the PoS interface statistics are inspected to check the card is locked to the data stream.

```
dag@endace:~$ dagsix -d dag0 -si
```

The tool displays a number of status bits that have occurred since last reading. The following example shows the interval is set to one second via the `-i` option.

`RAI`	Receive Alarm Indication. The optics report a receive error. One or more of the following two bits will also be set.
`RLE`	Receive Lock Error. The optics report a failure in clock recovery from the received signal.
`RPA`	Receive Power Alarm. The optics report insufficient optical input power (<-30dBm).
`LOS`	Loss Of Signal. The framer reports there is either no signal at receiver, or optical signal strength is too low to be recognized.
`LOC`	Loss Of Clock. The framer is not receiving a valid clock from the optics.
`OOF`	Out Of Frame. The framer is not locked to the SONET frame stream.
`LOF`	Loss Of Frame. The framer has asserted OOF for more than 3 milliseconds.
`LOP`	Loss Of Pointer. The framer cannot lock to the SONET/SDH frame pointers.
`FCS_ERR`	Number of PoS FCS [CRC32] errors since last reading.
`GOOD_PACKET`	Number of PoS frames received since last reading.
`RXF`	Receive Fifo Errors. Framer receive FIFO errors since last reading.

Example

The following is an example of a card locked to a PoS OC-192c stream carrying no traffic load:

```
dag@endace:~$ dagsix -d dag0 -si

 RAI   RLE   RPA   LOS   LOC   OOF  LOF   LOP  FCS_ERR  POS_PACKET  RXF
   0     0     0     0     0     0    0     0  4194303           0  255
   0     0     0     0     0     0    0     0        0           0    0
   0     0     0     0     0     0    0     0        0           0    0
   0     0     0     0     0     0    0     0        0           0    0
```

NOTE: The first second has high values as the counters have accumulated their values over more than one second and usually include pre-setup confirmation.

Extended statistics

Extended statistics are also available.

```
dag@endace:~$  dagsix -d dag0 -ei
```

```
RAI   RLE   RPA   LOS   LOC   OOF   LOF   LOP   FCS_ERR   GOOD_PACKET   RXF   BIPI   BIP2   BIP3   C2   RX_PARITY   TEMP
0     0     0     0     0     0     0     0     0         0             0     0      0      0      16   0           44
0     0     0     0     0     0     0     0     0         0             0     0      0      0      16   0           44
0     0     0     0     0     0     0     0     0         0             0     0      0      0      16   0           44
0     0     0     0     0     0     0     0     0         0             0     0      0      0      16   0           44
```

Extra counters

The following extra counters are available with the extended statistics option:

BIP1	Bit Interleaved Parity 1. SONET/SDH Section parity error count.
BIP2	Bit Interleaved Parity 2. SONET/SDH Line parity error count.
BIP3	Bit Interleaved Parity 3. SONET/SDH Path parity error count.
C2	Reflects content of SONET/SDH C2 overhead octet, or Path Signal Label. Typical settings are as follows: 16 PoS CF Cisco HDLC
RX_PARITY	Receive parity error count between the framer and receive FPGA.
TXF	Transmit Fifo Errors. Framer transmit FIFO errors since last reading.

Optical light levels

The following situation indicates a problem with optical light levels.

```
dag@endace:~$ dagsix -d dag0 -si

RAI   RLE   RPA   LOS   LOC   OOF   LOF   LOP   FCS_ERR   GOOD_PACKET   RXF
1     1     1     0     0     1     1     1     0         0             0
1     1     1     0     0     1     1     1     0         0             0
1     1     1     0     0     1     1     1     0         0             0
1     1     1     0     0     1     1     1     0         0             0
```

Although no signal is present, RPA is high, LOS and LOC may not be asserted. This can occur if the optics module outputs random noise when no input is present. In this case RPA, the Framing [LOF, OOF] and Pointer [LOP] errors can still be used to detect an error condition.

Correct configuration

In order to correct the configuration, proceed as follows:

- Ensure RAI, RLE and RPA (first three columns) are zero, check light levels
- Ensure no bip errors occur, otherwise check cabling and light levels
- Ensure the scrambling and CRC settings are ok

Little or no data information

On Packet-over-SONET [PoS] links it can happen that there is very little or no data information received. This typically indicates incorrect scrambling settings. While a default that matches typical link settings is provided, the actual configuration varies from network to network.

In this situation, vary the pscramble option and retry.

4.7 Inspect 10G Ethernet Interface Statistics

Description

Following inspection of the DAG 6.2S card PoS interface statistics the 10G Ethernet interface statistics are inspected to check the card is locked to the data stream.

```
dag@endace:~$ dagsix -d dag0 -si
```

The tool displays a number of status bits that have occurred since last reading. The following example shows the interval is set to one second via the `-i` option.

For LAN mode, the following statistics are available.

LAN Statistics

`RAI`	Receive Alarm Indication. The optics report a receive error. One or more of the following two bits will also be set.
`RLE`	Receive Lock Error. The optics report a failure in clock recovery from the received signal.
`RPA`	Receive Power Alarm. The optics report insufficient optical input power (<-30dBm).
`LOS`	Loss Of Signal. The framer reports there is either no signal at receiver, or optical signal strength is too low to be recognized.
`LOC`	Loss Of Clock. The framer is not receiving a valid clock from the optics.
`LOF`	Loss Of Frame. The framer has asserted OOF for more than 3 milliseconds.
`BER`	High Bit Error Rate detected, check optical level.
`LFT`	Local Fault, signal from peer is not being received correctly.
`RFT`	Remote Fault, peer is not receiving a signal correctly.
`FCS_ERR`	Number of Ethernet FCS errors since last reading.
`BAD_PACKET`	Number of errored packets received since last reading.
`GOOD_PACKET`	Number of correct packets received since last reading.
`RXF`	Receive Fifo Errors. Framer receive FIFO errors since last reading.

WAN Statistics For WAN mode, the following statistics are available.

RAI	Receive Alarm Indication. The optics report a receive error. One or more of the following two bits will also be set.
RLE	Receive Lock Error. The optics report a failure in clock recovery from the received signal.
RPA	Receive Power Alarm. The optics report insufficient optical input power (<-30dBm).
LOS	Loss Of Signal. The framer reports there is either no signal at receiver, or optical signal strength is too low to be recognized.
LOC	Loss Of Clock. The framer is not receiving a valid clock from the optics.
OOF	Out Of Frame. The framer is not locked to the SONET stream.
LOF	Loss Of Frame. The framer has asserted OOF for more than 3 milliseconds.
LOP	Loss Of Pointer. The framer cannot find the SONET/SDH frame pointers.
LOF	Loss Of Frame. The framer has asserted OOF for more than 3 milliseconds.
BER	High Bit Error Rate detected, check optical level.
LFT	Local Fault, signal from peer is not being received correctly.
RFT	Remote Fault, peer is not receiving a signal correctly.
BER	High Bit Error Rate detected, check optical level.
LFT	Local Fault, signal from peer is not being received correctly.
RFT	Remote Fault, peer is not receiving a signal correctly.
FCS_ERR	Number of Ethernet FCS errors since last reading.
BAD_PACKET	Number of errored packets received since last reading.
GOOD_PACKET	Number of correct packets received since last reading.
RXF	Receive Fifo Errors. Number of Framer receive FIFO errors since last reading.

Example The following is an example of a card locked to a 10G Ethernet LAN stream carrying no traffic load:

```
dag@endace:~$ dagsix -d dag0 -si

RAI RLE RPA LOS LOC LOF BER LFT RFT FCS_ERR BAD_PACKET GOOD_PACKET RXF
  0   0   0   0   0   0   0   0   0       0          0     9090612   0
  0   0   0   0   0   0   0   0   0       0          0           0   0
  0   0   0   0   0   0   0   0   0       0          0           0   0
  0   0   0   0   0   0   0   0   0       0          0           0   0
  0   0   0   0   0   0   0   0   0       0          0           0   0
```

Example The following is an example of a card locked to a 10G Ethernet WAN stream carrying no traffic load:

```
dag@endace:~$ dagsix -d dag0 -si

RAI RLE RPA LOS LOC OOF LOF LOP LOF BER LFT RFT FCS_ERR BAD_PACKET GOOD_PACKET RXF
  0   0   0   0   0   0   0   0   0   0   0   0       0          0     5458160   0
  0   0   0   0   0   0   0   0   0   0   0   0       0          0           0   0
  0   0   0   0   0   0   0   0   0   0   0   0       0          0           0   0
  0   0   0   0   0   0   0   0   0   0   0   0       0          0           0   0
```

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

LAN Extended statistics Extended statistics are also available. The following example shows extended statistics for the above LAN configuration.

```
dag@endace:~$  dagsix -d dag0 -ei

RAI RLE RPA LOS LOC LOF BER LFT RFT FCS_ERR BAD_PACKET GOOD_PACKET RXF RX_PARITY TEMP
  0   0   0   0   0   0   0   0   0       0          0    64057561   0         0   67
  0   0   0   0   0   0   0   0   0       0          0    14629029   0         0   67
  0   0   0   0   0   0   0   0   0       0          0    14629444   0         0   67
  0   0   0   0   0   0   0   0   0       0          0    14629463   0         0   67
```

LAN Extra counters The following extra counters are available with the extended statistics option:

RX_PARITY	Receive parity error count between the framer and receive FPGA.
TEMP	Temperature of Rx FPGA in degrees Celsius.

WAN Extended statistics

Extended statistics are also available. The following example shows extended statistics for the above WAN configuration.

```
dag@endace:~$  dagsix -d dag0 -ei
```

```
RAI RLE RPA LOS LOC OOF LOF LOP LOF BER LFT RFT FCS_ERR BAD_PACKET GOOD_PACKET RXF  BIP1     BIP2  BIP3 C2 RX_PARITY TEMP
  1   1   0   0   0   1   1   1   1   0   1   0       1          1  1114253956   0 65535    65535 65535 01         0   49
  1   1   0   0   0   1   1   1   1   0   1   0       0          0           0   0 32885    24725 24725 01         0   49
  1   1   0   0   0   1   1   1   1   0   1   0       0          0           0   0 32994    24717 24717 01         0   49
  1   1   0   0   0   1   1   1   1   0   1   0       0          0           0   0 33056    24711 24711 01         0   49
  1   1   0   0   0   1   1   1   1   0   1   0       0          0           0   0 32991    24727 24727 01         0   50
```

Extra counters

The following extra counters are available with the extended statistics option:

BIP1	Bit Interleaved Parity 1. SONET/SDH Section parity error count.
BIP2	Bit Interleaved Parity 2. SONET/SDH Line parity error count.
BIP3	Bit Interleaved Parity 3. SONET/SDH Line parity error count.
C2	Reflects content of SONET/SDH C2 overhead octet, or Path Signal Label. Typical settings are as follows: 16 PoS CF Cisco HDLC
RX_PARITY	Receive parity error count between the framer and receive FPGA.
TEMP	Temperature of Rx FPGA in degrees Celsius.

Optical light levels in LAN mode

The following situation indicates a problem with optical light levels in LAN mode.

```
dag@endace:~$ dagsix -d dag0 -si
```

```
RAI RLE RPA LOS LOC LOF BER LFT RFT FCS_ERR BAD_PACKET GOOD_PACKET RXF RX_PARITY TEMP
  1   1   1   0   0   1   0   1   0       0          0           0   0         0   54
  1   1   1   0   0   1   0   1   0       0          0           0   0         0   54
  1   1   1   0   0   1   0   1   0       0          0           0   0         0   54
  1   1   1   0   0   1   0   1   0       0          0           0   0         0   54
```

Optical light levels in WAN mode

The following situation indicates a problem with optical light levels in WAN mode.

```
dag@endace:~$ dagsix -d dag0 -si
```

```
RAI RLE RPA LOS LOC OOF LOF LOP LOF BER LFT RFT FCS_ERR BAD_PACKET GOOD_PACKET RXF  BIP1     BIP2  BIP3 C2 RX_PARITY TEMP
  1   1   1   0   0   1   1   1   1   0   1   0       0          0           0   0 65535    65535 65535 01         0   52
  1   1   1   0   0   1   1   1   1   0   1   0       0          0           0   0 32972    24769 24769 01         0   52
  1   1   1   0   0   1   1   1   1   0   1   0       0          0           0   0 33099    24755 24755 01         0   52
  1   1   1   0   0   1   1   1   1   0   1   0       0          0           0   0 33120    24767 24767 01         0   52
```

Although no signal is present, RPA is high, LOS and LOC may not be asserted. This can occur if the optics module outputs random noise when no input is present. In this case RPA, the Framing [LOF, OOF] and Pointer [LOP] errors can still be used to detect an error condition.

Correct configuration

In order to correct the configuration, proceed as follows:

- Ensure RAI, RLE and RPA (first three columns) are zero, check light levels
- Ensure no bip errors occur, otherwise check cabling and light levels
- Ensure the scrambling and CRC settings are ok

Little or no data information

On WAN links it can happen that there is very little or no data information received. This typically indicates incorrect scrambling settings. While a default that matches typical link settings is provided, the actual configuration varies from network to network.

In this situation, vary the `pscramble` option and retry.

4.8 Reporting Problems

Description

If there are unresolved problems with a DAG card or supplied software, contact Endace Technical Support via the email address support@endace.com. Supplying sufficient information in an email enables effective response.

Problem checklist

The exact information available to users for trouble, cause and correction analysis may be limited by nature of the problem. The following items assist a quick problem resolution:

Ref	Item
1.	DAG card[s] model and serial number.
2.	Host PC type and configuration.
3.	Host PC operating system version.
4.	DAG software version package in use.
5.	Any compiler errors or warnings when building DAG driver or tools.
6.	For UNIX/FreeBSD users only: 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: `dagsix`
10.	Network link statistics reported by: `dagsix -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.

Chapter 5: Running Data Capture Software

Introduction For a typical measurement session, ensure the driver is loaded, the firmware has been downloaded, and the card has been configured.

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

- Starting Data Capture Session
- High Load Performance

5.1 Starting Data Capture Session

Description Starting the capture of data involves a typical measurement session, the card operating in variable length mode, starting and stopping the session.

Process The following process describes starting a data capture session.

Process	Description

Typical measurement session.	For a typical measurement session, , load the driver, then load the FPGA images to each DAG card. For example:
	
	`/dagload`
	`/dagrom -rvp -d dag0 < xilinx/dag62pcix-erf.bit`
	`/dagld -x -d dag0 xilinx/dag62rx-erf.bit`
	
	Then set, and check the integrity of the cards physical layer. This process is described in Chapter 4 of this document.
	
	`/dagsix -d dag0 default`

Process	Description
Variable length mode.	The DAG 6.2S always operates in variable-length mode. The number of bytes collected from the packet is configured with `slen`. This defaults to 48, and can range from 48 to 2040, and be a multiple of 8. Packets longer than `slen` will be truncated to slen. Packets shorter than `slen` produce shorter records, saving bandwidth and storage space. For example, for full packet capture: `/dagsix –d dag0 slen=1552` Values of `slen` higher than the default may lead to increased packet loss during captures under high link load, due to limited PCI-X bandwidth.
Starting a capture session.	A capture session is started as follows: `/dagsnap –d dag0 –v –o tracefile0` The option `–v` is used to provide user information during capture; this can be omitted for automated trace runs. If the `tracefile` parameter is not specified the tool will write to stdout, which can be used to pipeline `dagsnap` with other tools from the `dagtools` package.
Stopping `dagsnap` running	By default `dagsnap` will run forever. `dagsnap` can be stopped with a signal: `killall dagsnap` `dagsnap` can also be configured to run for a fixed number of seconds and then exit with the `–s` option.

5.2 High Load Performance

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.

Avoiding packet loss

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

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 `log /var/log/messages`.

If the user process is writing records to hard disk, it may be necessary to use a faster disk or disk array. If records are being processed in real-time, a faster host CPU may be required.

Increasing buffer size

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

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

128MB or more is suggested for Linux/FreeBSD.

For the DAG 6.2S card Windows operating system the upper limit is 256MB.

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.

Chapter 6: Synchronizing Clock Time

Description

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

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

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

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

DUCK configuration

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

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

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

Common synchronization

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

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

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

More information is on the Endace website, http://www.endace.com/accessories.htm, or the TDS 2/TDS 6 Units Installation Manual.

In this chapter

This chapter covers the following sections of information.

- Configuration Tool Usage
- Time Synchronization Configurations
- Synchronization Connector Pin-outs

6.1 Configuration Tool Usage

Description

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

Synchronization settings are controlled by the `dagclock` utility.

Example

```
dag@endace:~$ dagclock -h
Usage: dagclock [-hvVxk] [-d dag] [-K <timeout>] [-l
<threshold>] [option]

     -h  --help,--usage   this page
     -v  --verbose        increase verbosity
     -V  --version        display version information
     -x  --clearstats     clear clock statistics
     -k  --sync           wait for duck to sync before
                          exiting
     -d  dag              DAG device to use
     -K  timeout          sync timeout in seconds, default
                          60
     -l  threshold        health threshold in ns, default
                          596
  Option:
        default           RS422 in, none out
        none              None in, none out
        rs422in           RS422 input
        hostin            Host input (unused)
        overin            Internal input (synchronize to
                          host clock)
        auxin             Aux input (unused)
        rs422out          Output the rs422 input signal
        loop              Output the selected input
        hostout           Output from host (unused)
        overout           Internal output (master card)
        set               Set DAG clock to PC clock
        reset             Full clock reset.  Load time
                          from PC, set rs422in, none out
```

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

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

6.2 Time Synchronization Configurations

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

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

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

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

6.2.1 Single Card no Reference Time Synchronization

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

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

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

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

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

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

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

6.2.2 Two Cards no Reference Time Synchronization

Description

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

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

Synchronizing cards

If both cards are to be accurately synchronized, but not so for absolute time of packet time-stamps being correct, then one card is configured as the clock master for the other.

Locking cards together

Although the master card's clock will drift against UTC, the cards are locked together.

The cards are locked together by connecting the 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.

Preventing time-stamps drift

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

The cards are locked together by connecting the synchronization connector ports of both cards with a standard RJ-45 Ethernet cross-over cable.

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

```
dag@endace:~$ dagclock -d dag0 none overin overout
muxin   over
muxout  over
status  Synchronized Threshold 11921ns Failures 0 Resyncs
0
error   Freq -691ppb Phase -394ns Worst Freq 143377ppb
Worst Phase 88424ns
crystal Actual 49999354Hz Synthesized 16777216Hz
input   Total 87464 Bad 0 Singles Missed 0 Longest
Sequence Missed 0
start   Wed Apr 27 14:27:41 2005
host    Thu Apr 28 14:59:14 2005
dag     Thu Apr 28 14:59:14 2005
```

The slave card configuration is not shown, the default configuration is sufficient.

6.2.3 Card with Reference Time Synchronization

Description

The best timestamp accuracy occurs when 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.

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.

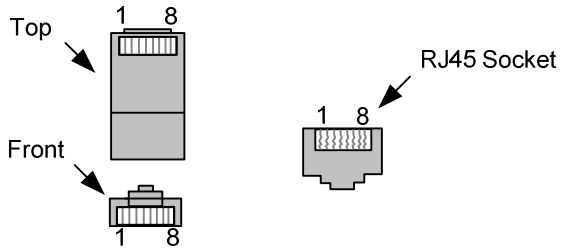

Figure 6-1. RJ45 Plug and Socket Connector Pin-outs.

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.

Chapter 7: Data Formats

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

- Data Formats
- Timestamps

7.1 Data Formats

Description The DAG 6.2S uses the ERF Type 1 POS HDLC Variable Length Record and the ERF Type 2 Ethernet Variable Length Record. Timestamps are in little-endian [Pentium native] byte order. All other fields are in big-endian [network] byte order. All payload data is captured as a byte stream, no byte re-ordering is applied.

Table Table 7-1 shows the generic variable length record. The diagram is not to scale.

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

Table 7-1. Generic Variable Length Record.

Data format The following is an overview of the data format used.

Data Format	Description
type:	This field contains an enumeration of the frame subtype. If the type is zero, then this is a legacy format. 0: TYPE_LEGACY 1: TYPE_HDLC_POS: PoS w/HDLC framing 2: TYPE_ETH: Ethernet 3: TYPE_ATM: ATM Cell 4: TYPE_AAL5: reassembled AAL5 frame 5: TYPE_MC_HDLC: Multi-channel HDLC frame 6: TYPE_MC_RAW: Multi-channel Raw link data 7: TYPE_MC_ATM: Multi-channel ATM Cell

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 [internal error] 7-6: reserved
Rlen: record length	Total length of the record transferred over PCI bus to storage.
Lctr: *loss counter*	A 16 bit counter, recording the number of packets lost since the previous record. Records can be lost between the DAG card and memory hole due to overloading on PCI bus. The counter starts at zero, and sticks at 0xffff.
Wlen: *wire length*	Packet length including some protocol overhead. The exact interpretation of this quantity depends on physical medium.

Table Table 7-2 shows the Type 1 POS HDLC variable length record. The diagram is not to scale.

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

Table 7-2. Type 1 POS HDLC Variable Length Record.

Table Table 7-3 shows the Type 2 Ethernet variable length record. The diagram is not to scale.

timestamp		
timestamp		
type:2	flags	rlen
lctr		wlen
offset	pad	rlen-18
bytes of frame		

Table 7-3. 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 in the Winsock library.

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

Description,continued

Example codes Here is some example code showing how a 64-bit ERF timestamp (erfts) can be converted into a struct timeval representation (tv).

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

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