CoaXPress Frame Grabbers

Aon, Claxon & Cyton

Hardware Reference Manual

BitFlow, Inc.
400 West Cummings Park, Suite 5050
Woburn, MA 01801
USA
Tel: 781-932-2900
Fax: 781-933-9965
Email: support@bitflow.com
Web: www.bitflow.com
Revision C.1

Revision History:

Revision	Date	Comments
Pre	2013-10-10	Preliminary
A.0	2014-10-10	Synchronized with SDK 5.90
A.1	2016-07-04	Synchronized with SDK 6.00
A.2	2016-04-05	Synchronized with SDK 6.20
B.0	2017-04-01	Added support for the Aon-CXP, Synchronized with SDK 6.30
B.1	2017-12-16	Minor update and corrections
C.0	2020-07-01	Added Claxon, Synchronized with SDK 6.50
C.1	2023-08-01	Added Claxon-Fiber, Synchronized with SDK 6.6

Table of Contents

3 - The StreamSync Buffer Manager

4 - Timing Sequencer

5 - The Aon, Axion, Claxon and Cyton I/O System

6 - The I/O System Registers

7 - Encoder Divider

8 - Quadrature Encoder

9 - Quadrature Encoder and Divider Registers

10 - CXP Subsystem Registers

11 - Encoder Based Processing Control

12 - CXP 2.0 Registers

13 - Specifications

14 - Mechanical

Preface

Chapter P

P.1 Purpose

This Hardware Reference Manual is intended for anyone using the Aon, Cyton or Claxon CoaXPress (CXP) frame grabbers. The purpose of this manual is two-fold. First, this manual completely describes how the board works. Second, it is a reference manual describing in detail the functionality of all of the board's registers.

Note: The Aon, Claxon and Cyton are all architecturally very similar. The major differences are the number of links and VFGs, support for CXP-12 and support for CXP over Fiber (CoF). Because the Cyton came first, this manual will use the work "Cyton" generically for all BitFlow CXP frame grabbers. In cases where the products differ, the distinction will be made as to which features are on which product family.

P.1.1 Support Services

BitFlow, Inc. provides both sales and technical support for the Cyton and Aon families of products.

P.1.2 Technical Support

Our web site is www.bitflow.com.

Technical support is available at 781-932-2900 from 9:00 AM to 6:00 PM Eastern Standard Time, Monday through Friday.

For technical support by email (support@bitflow.com) please include the following:

Product name
Camera type and mode being used
Output from VerCheck
Computer CPU type, PCI chipset, bus speed
Operating system
Example code (if applicable)

P.1.3 Sales Support

Contact your local BitFlow Sales Representative, Dealer, or Distributor for information about how BitFlow can help you solve your most demanding camera interfacing problems. Refer to the BitFlow, Inc. web site (www.bitflow.com) for a list of North American representatives and worldwide distributors.

P.1.4 Conventions

Table P-1 shows the conventions that are used for numerical notation in this manual.

Table P-1 Base Abbreviations

Base	Designator	Example
Binary	b	1010b
Decimal	None	4223
Hexidecimal	h	12fah

Table P-2 shows the numerical abbreviations that are used in this manual.

Table P-2 Numeric Abbreviations

Abbreviation	Value	Example
K	1024	256K
M	1048576	1M

P.2 Bitfield definitions

P.2.1 Example Bitfield Definition

Here is what each bitfield definition looks like:

BITFIELD R/W, CON0[7..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Bitfield discussion.

P.2.2 Bitfield Definition Explanation.

The definitions is broken into three sections (see Table P-3).

Table P-3 Bitfield Sections.

Section	Meaning
Bitfield name	This is the name of the bitfield. This name is use to program this bitfield from software or from within and camera configuration file. When programming bitfields from software using a Peek or Poke function, the bitfield is preceded with "REG_". For example the bitfield CFREQ is referred to in software as REG_CFREQ.
Bitfield details	This section describes how the bitfield is accessed. The first part describes the how the bits can be accessed. For example R/W means the register can be both read and writen. See theTable P-4 for details.The second part is the wide register that the bitfield is located in. In the example above this bitfield is in CON0. Following the wide register name is a bitfield location description, in hardware engineering format. For example, [7..0], means the bitfield has 8 bits, location in positions 0 to 7. Finally this section also indicates if the register is specific to only one product family.
Bitfield discussion	This section explains the purposed of the bitfield in detail. Usually meaning of every possible value of the bitfield is listed.

Table P-4 explains the abbreviations used in the bitfield definitions.

Table P-4 Abbreviations

Access	Meaning
R/W	Bitfield can be read and written.
RO	Bitfield can only be read. Writing to this bit has no effect.
WO	Bitfield can only be written. Reading from this bit will return meaningless values.
Karbon-CL	This bitfield is functional on the Karbon-CL.
Neon	This bitfield is functional on the Neon
Aon-CXP	This bitfield is functional on the Aon-CL family
Claxon-CXP	This bitfield is functional on the Claxon-CXP family
Claxon-FXP	This bitfield is functional on the Claxon-Fiber (CoaXPress over Fiber)
Cyton-CXP	This bitfield is functional on the Cyton-CXP family
Axion-CL	This bitfield is functional on the Axion-CL family

General Description and Architecture

Chapter 1

1.1 The Aon, Claxon and Cyton families

The purpose of this chapter is to explain, at a block diagram level, how the Aon, Claxon and Cyton work. Here is a list of main models:

AON-PC2-CXP1, provides one 6.25 Gb/S CXP link
CLX-PC3-CXP2, provides two 12.5 Gb/S CXP links
CLX-PC2-CXP1, provides one12.5 Gb/S CXP link
CLX-PC3-CXP4, provides four12.5 Gb/S CXP links
CLX-PC3-FXP4, provides four12.5 Gb/S CoaXPress over Fiber links
CYT-PC2-CXP4, provides four 6.25 Gb/S CXP links
CYT-PC2-CXP2, provides two 6.25 Gb/S CXP links

1.1.1 CoaXPress

In order to understand how the Aon, Claxon and Cyton work, it is helpful to understand the basics of CoaXPress. It is beyond the scope of this manual to describe how CoaXPress works, however, more information on the CoaXPress specification is available from www.coaxpress.com.

1.1.2 CoaXPress over Fiber

CoaXPress over Fiber is an extension of the CoaXPress specification. This extension provides a method for carrying CoaXPress packets over fiber optic cables. One of them main design goals of this specification was to support existing CoaXPress encode/decode engines. This was deemed the best path to support existing firms quickly releasing fiber products. In addition, this would also existing customer to easily switch to fiber with almost no learning curve.

BitFlow believes these goals have largely been achieved. Customers can switch between the copper and fiber versions of the Claxon with no changes to their software. All concepts learning from using the copper Claxon/Cyton apply directly to the Claxon-Fiber; all the BitFlow tools and utilities are the same. There really is no learning curve.

1.1.3 Virtual vs. Hardware

It's important to understand how this manual works. Some chapters of this manual discuss the Cyton-CXP as a hardware platform (this chapter is a good example). While other chapters discuss the details of the virtual frame grabbers (VFG) that this hard-

ware platform supports. The concept of the virtual frame grabber is described below, but basically the idea is that one hardware platform can support more than one device. In the case of the Aon, Claxon and Cyton , these devices are frame grabbers.

Note that we are not using the word virtual here in the sense of "a software virtualization of a hardware device", these VFGs are real hardware. The reason we using "virtual" is because the term "frame grabber" has more than one meaning. It can mean the piece of hardware that you put in your computer, or it can mean the device that the your software application is controlling and getting images from. For the purposes of this manual, "virtual frame grabber" means the device that your application is interfaces with. While this might sound complicated, the implementation is simple. You plug our Cyton-CXP frame grabber into your PC, and your application interacts with one or more VFGs available. Everything else is taken care of by the BitFlow drivers.

1.1.4 The Virtual Frame Grabber (VFG)

The idea behind the VFG is to separate the hardware platform (connectors, laminate, FPGAs, etc.) from the frame grabbing functionality that software applications work with. The primary reason behind this separation is that the turn around time for hardware is much longer than the turn around time for modifying virtual frame grabbers. To create a brand new virtual frame grabber, or to modify an existing one, simply requires writing new firmware or updating existing firmware.

The idea of modifying a frame grabber by making changes to its firmware is not new. BitFlow has been doing this since its very first product. However, unique to BitFlow products is the fact the entire frame grabber is written in firmware. The only fixed hardware components are the interfaces to the outside world (e.g. the interface chips on the front end). Everything else that makes up the board, camera control, data buffering, DMA engine, etc. is written in firmware. This gives the platform incredible levels of flexibility and opens the door to unlimited customization.

1.1.5 Configuration Spaces

The Aon, Claxon and Cyton models supports different numbers of VFGs. Each VFG appears to operating system and your software as a separate device. Each VFG will can connect to one or more CXP links. The following figures show the block diagrams for each possible configuration.

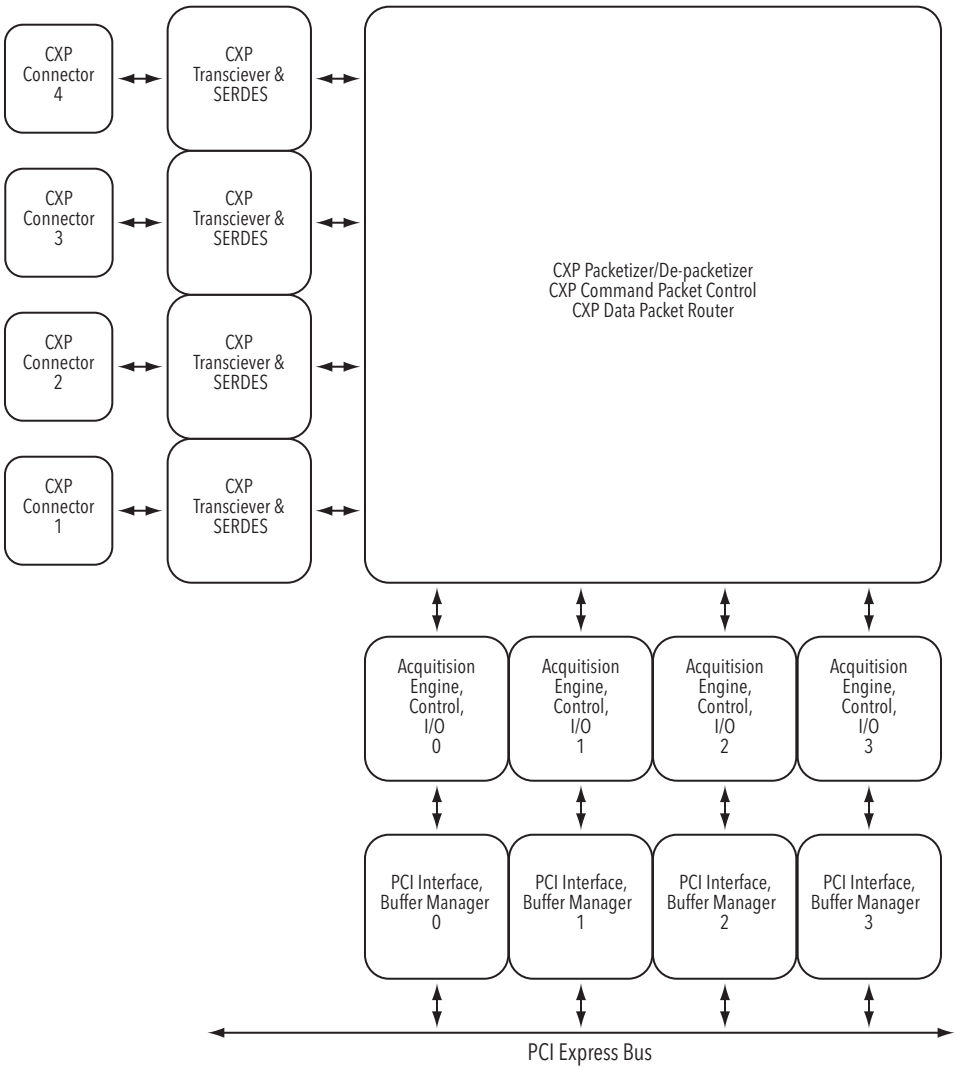

Figure 1-1 The Claxon-CXP4 and Cyton-CXP4 Block Diagram

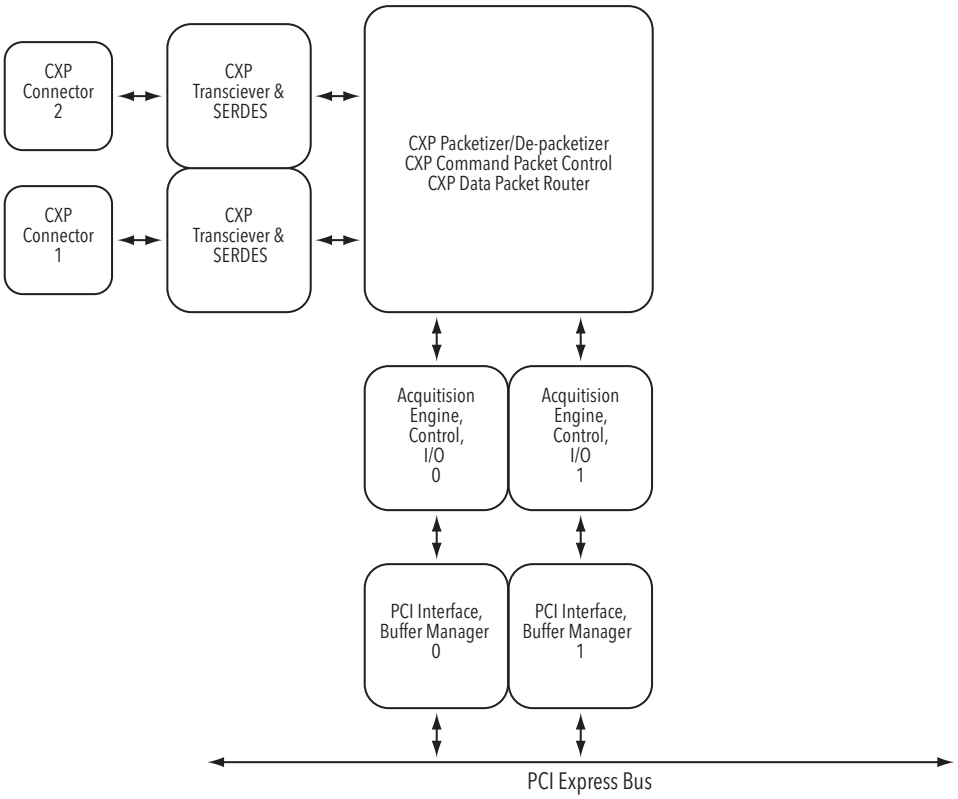

Figure 1-2 The Claxon-CXP2 and Cyton-CXP2 Block Diagram

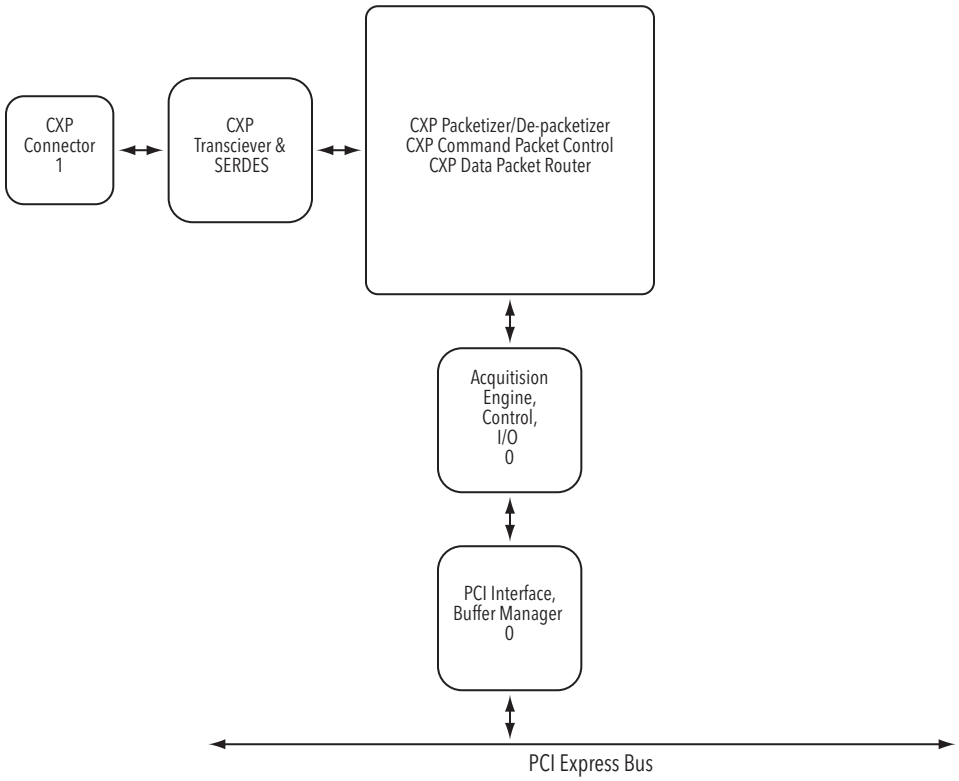

Figure 1-3 The Aon-CXP and Claxon-CXP1 Block Diagram

1.2 General Description

1.2.1 Aon

The Aon-CXP is an x2 PCI Express Gen2 board. It will work in a x4, x8 or x16 Gen2 or Gen3 slot.

The Aon-CXP has only one CXP link and only on VFG. This board only works with single link CXP cameras.

The Aon uses DIN 1.0/2.3 connectors.

The Aon-CXP1 support only one single-link camera.

1.2.2 Claxon

The Claxon-CXP2 and Claxon-CXP4 are a PCIe x8 Gen 3 boards. They will work in x1, x4, x8 ans x16 Gen2 or Gen3 slots. DMA performance will be reduce for x1 and x4 slots and/or Gen 2 slots.

The Claxon-CXP1 is a PCIe x4 Gen 2 board. It will work in x1, x4, x8 ans x16 Gen2 or Gen3 slots. DMA performance will be reduced in an x1 slot.

The Claxon uses HD-BNC, also known as micro BNC, connectors.

The Claxon-CXP1 support only one single-link camera. The Claxon-CXP2 supports one dual-link camera or two single-link cameras. The Claxon-CXP4 supports four single-link cameras, two dual-link cameras or one quad-link cameras.

1.2.3 Claxon-Fiber

The Claxon-Fiber is a PCIe x8 Gen 3 boards. It will work in x1, x4, x8 ans x16 Gen2 or Gen3 slots. DMA performance will be reduce for x1 and x4 slots and/or Gen 2 slots.

The Claxon-Fiber uses a QFSP+ “cage” which supports a wider variate of cable assemblies. While there are copper cable assemblies compatible with the Claxon-Fiber, the primary focus is on fiber optic cable assemblies. The Claxon-Fiber supports fiber cables with up to four fibers in and four fibers out. These can support one quad link CoaXPress over Fiber (CoF) camera, or four single link CoF cameras with the proper break out cable.

1.2.4 Cyton

The Cyton-CXP is an x8 PCIe Gen 2 board. It requires a PCIe x8 or x16 slot that is Gen 2 or Gen 3. The Cyton-CXP will not work PCIe x4 or Gen 1 slots.

The Cyton uses DIN 1.0/2.3 connectors.

The Cyton-CXP2 supports one dual-link camera or two single-link cameras. The Cyton-CXP4 supports four single-link cameras, two dual-link cameras or one quad-link cameras.

1.2.5 Single-Link Configuration

Figure 1-4 shows the block diagram of a VFG of any model when it used with a single link camera. Multi-link configurations are shown later in this chapter.

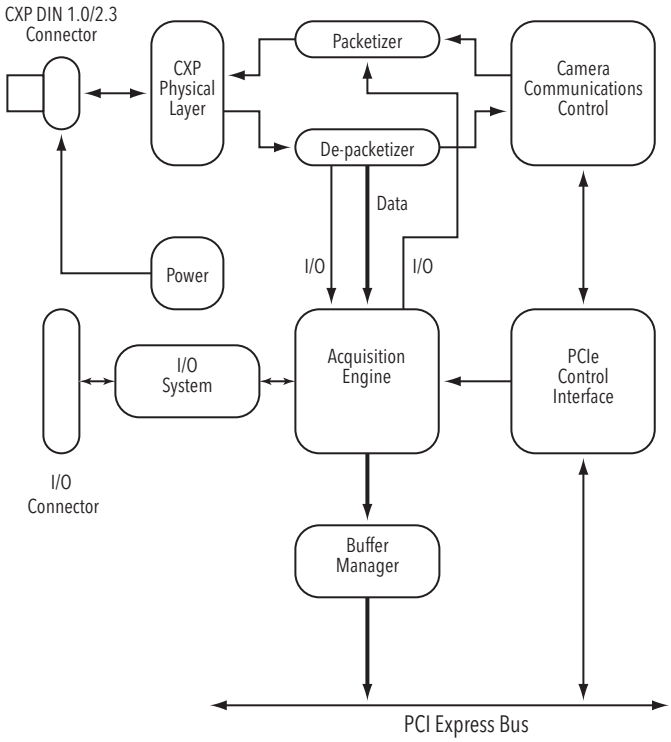

Figure 1-4 Single Link Configuration Block Diagram

1.2.6 Video Data

Three types of packets are sent from the camera on the CoaXPress downlink: Video (stream) data, control (command) packets and Trigger packets. Video packets contain video information (pixels). Control packets contain commands and response. The control packets are use to read/write from/to the cameras internal register space. Trigger packets are used to send triggers to the host.

Packets containing video data are sent to a stream assembler. The stream assembler builds up lines of video data which are then sent to the acquisition and control logic (see Figure 1-4). From this point on the Aon/Claxon/Cyton work very similar to all of

BitFlow's frame grabbers. The acquisition circuit determines which pixels of which lines of which frames are to be acquired. The correct pixels are then sent to a FIFO, the output of which read by the DMA engine. The DMA engine then DMAs the data to host memory.

CoaXPress cameras can output video on multiple links simultaneously. There is no correlation between parts of the sensor and CXP links (like there is between sensor taps and Camera Link taps). Any part of any line can be sent on any link in a multi-link camera. The stream assembler must decode the packet headers and assemble raster format lines.

1.2.7 Command and Control Packets

The second type of packet coming from the camera is a control packet. When the board sees a control packet, it unwraps the packet and sends the contents into a control data FIFO. The host then reads and decodes this FIFO. Currently (as of CoaXPress specification 1.1) control packets are synchronous. This means that the camera will only send a control packet in response to a control packet sent from the host. For this reason, software can be fully under control of the control packets sent from the camera. It is not necessary to have an interrupt driven control packet circuit as there is for serial communications in Camera Link. However, this may change with future revision os the CoaXPress specification.

1.2.8 CXP Triggers

The third type of packet the Aon/Claxon/Cyton handle are the trigger packets. The camera can send a trigger packet at any time. The Cyton-CXP maps the trigger packet into the board's trigger circuitry. It is one possible source for the internal trigger signal.

1.2.9 Uplink

CoaXPress provides a high speed uplink to control the camera. The uplink is also packet based like the downlink. However, the uplink only supports two types of packets, control and Trigger. These packets are similar to the packets in the downlink. For control packets, the BitFlow software builds up the packet in the board's outgoing control packet FIFO. Once the packet is fully built, a single bit sends the packet to the camera. Trigger packets work in a similar manner, the trigger packet is a destination for the board's I/O system, and any number of sources can be routed to the CXP trigger.

1.2.10 Cyton/Aon-CXP I/O system

The Cyton-CXP has a sophisticated I/O system, which is extremely flexible. The system take in many inputs, routes them to a number of internal signals which can be further manipulate, then routes the results to a wide rand of outputs. The I/O system is discuss in more detail in Section 7.1.

1.2.11 The Timing Sequencer Signal Generator

With the release of the Cyton-CXP, BitFlow introduced a new signal generator, the Timing Sequencer. The Timing Sequencer (TS) is more flexible and more power than the timing generators used on early BitFlow frame grabbers. It has the ability to output multiple different size pulses, each of which can free-run or require a trigger. The TS is more accurate than the NTG and has a finer granularity. The TS can also be changed on the fly, with switch overs to the new timing exactly synchronized. See section 4.1 for more information.

The Aon and Claxon provides the same TS as the Cyton.

1.2.12 The StreamSync System

The Cyton-CXP has a brand new, designed from scratch acquisition and DMA engine called the StreamSync system. The StreamSync system has been designed to optimize acquisition and DMA throughput over the PCIe bus given a wider variety of internal PC conditions. In addition, the Stream Sync system has been designed to automatically resync and recover should there every be packet lost (either on the input or the output side of the board), resulting in much more usable and fault tolerant image sequences in host memory. For more information see Section 2.1 and Section 3.1.

The Aon and Claxon use the same StreamSync Acquisition engine as the Cyton.

1.2.13 The Volume Of Interest Acquisition Engine

The Cyton-CXP introduces the concept of Volume of Interest (VOI) as part of its StreamSync Acquisition Engine. This has been designed from the ground up to satisfy the needs of real world machine vision application. The VOI provide robust and flexible programming that can handle of a wide variety of pixel, line, frame and sequence acquisition commands either manually from software control, or externally via hardware triggers. There is fully support for X and Y offsets, X and Y Region of interests, sequences and sequences of sequences. See section 2.2 for more information.

The Aon and Claxon also use the VOI Acquisition Engine.

1.2.14 CoaXPress Power

The CoaXPress specification specifies that the frame grabber must be cable of supplying up to 13 W at 24V on each CXP link. The Aon/Claxon/Cyton conforms to this specification. Some cameras do not require power, so the Aon/Claxon/Cyton can optionally turn power on or off via its registers. Normally this information is part of the camera configuration file, thus files for cameras that require power are so indicated.

The Aon/Claxon/Cyton automatically powers up all links that need power (i.e. correctly respond to the sense circuit). This happens as soon as the system is booted.

The Aon/Claxon/Cyton constantly monitors the current on each CXP link, if either over current or under current conditions exist, the power will be turned off. The monitoring system is purely in hardware, so no host computer intervention is required in order to safeguard the power source.

For situation where the camera requires more power than the PCIe bus can supply to the frame grabber, the P4 connector can be use. This connector can be connect to the PC's power supply and all camera power will come from this connector.

Note: If the total amount of all cameras connected to the frame grabber exceeds 15 Watts, then the auxiliary power connector must be used. For example, if four single link cameras each taking 4 watts are connected, then auxiliary power should be used. Similarly, if a single camera quad link camera that takes 20 wats on to links is connected, then auxiliary power must be used.

Note: Because the Aon-CXP has only one link, there is no need for auxiliary power. The Aon-CXP can derive enough power from the PCIe bus to power one CXP link.

Note: The Claxon-Fiber does not provide power to a connected camera.

1.3 Firmware

Unlike many of BitFlow's previous models of frame grabbers, the Aon/Claxon/Cyton do not swap firmware on the fly. These boards are shipped with firmware that supports the latest CoaXPress Specification and has been tested with all known cameras at the time of the release. However new features may be added and anomalies corrected from time to time. This updates will take the form of a new firmware file (*.fsh). You may receive an updated firmware file as part of support issue, or a new firmware release may be part of a new SDK. In general, it is best to update the firmware on your board whenever you upgrade to a major new version of the SDK.

To update the board's firmware, type to the following command in a console window:

FWdownload

follow the instruction of the download program.

Note: After the firmware download process is complete, you must power down your computer in order for the new firmware to become active. Rebooting (AKA Restart, Warm Reboot) is not sufficient to force the new firmware to be downloaded. Only a cold reboot will work.

1.4 Aon/Claxon/Cyton Camera Configuration Files

The Cyton is the first member of BitFlow's Gen 2 family. This frame grabber, like all Gen 2 boards use and XML based camera configuration file. This differs from previous models of BitFlow's frame grabbers that have all used a binary proprietary file format (which mean they could only be edited using BitFlow's tools). The file format uses the extension "BFML" but is actually an XML file with an XML compliant schema. The schema file is installed automatically and is called "BFML.xsd".

The Aon and Claxon uses the same format camera configuration file as the Cyton-CXP. A BFML configuration file for a single link camera will work with the Aon, Claxon and the Cyton. Multi-link BFML files require a model with the required number of links. For example, the a quad-link camera would require the Claxon-CXP4 or the Cyton-CXP4

Starting with release 6.30, the SDK shipped with a dedicated BFML file editor called CamML. This application makes modifying BFML files very easy and intuitive. In addition, CamML can open a display window and show video from the camera/frame grabber configured with the current configuration.

In addition, BFML files can be edited in any text editor. User's familiar with XML will understand the format right away. Users not familiar with XML files should not have too much trouble editing the files, but the XML file format is used everywhere and there are many resources available for learning the format.

The BFML file format is documented on BitFlow's website. Please see the downloads page for a link to the BFML documentation.

Note: The tools used to edit previous BitFlow camera configuration files (CamEd, CamVert) can not be used to edit BFML files.

1.4.1 BFML Camera File Modes

Previous BitFlow camera configuration file only supported a single mode of camera operation. In order to support multiple (e.g. free-running, one-shot, triggered, encoder, etc.) modes for a given camera file, multiple files were required, one for each mode. The BFML file format can contain an unlimited number of camera modes. This makes things much simpler since only one file is needed for each model of camera. Then the different modes of operation are contained within that one BFML file.

Switching between camera modes is easy, this can be done via SysReg, where each of the camera modes are enumerated, and the user can pick which mode they want to use. The modes are also available from the API. There are function to enumerate the modes and function to switch modes on the fly.

Customer can create their own modes as they see fit. They can simple copy an existing mode and change to suite there needs. The new mode should have a new name (also the comment should be updated). Once this file is save, the mode will be available in SysReg as well as from the API.

1.5 The Routing of CXP Links to VFGs on the muli-link Cyton and Claxon

What Figure 1-4 above does not show is how the CXP link in this case is routed to the VFG (ie. PCI space and acquisition engine). As hinted at in Figure 1-1, the board has quite a bit of flexibility in this area. Each VFG has its own acquisition and control circuit which can be connected to any of the available CXP link. Figure 1-5 shows the all the theoretical routing possibilities.

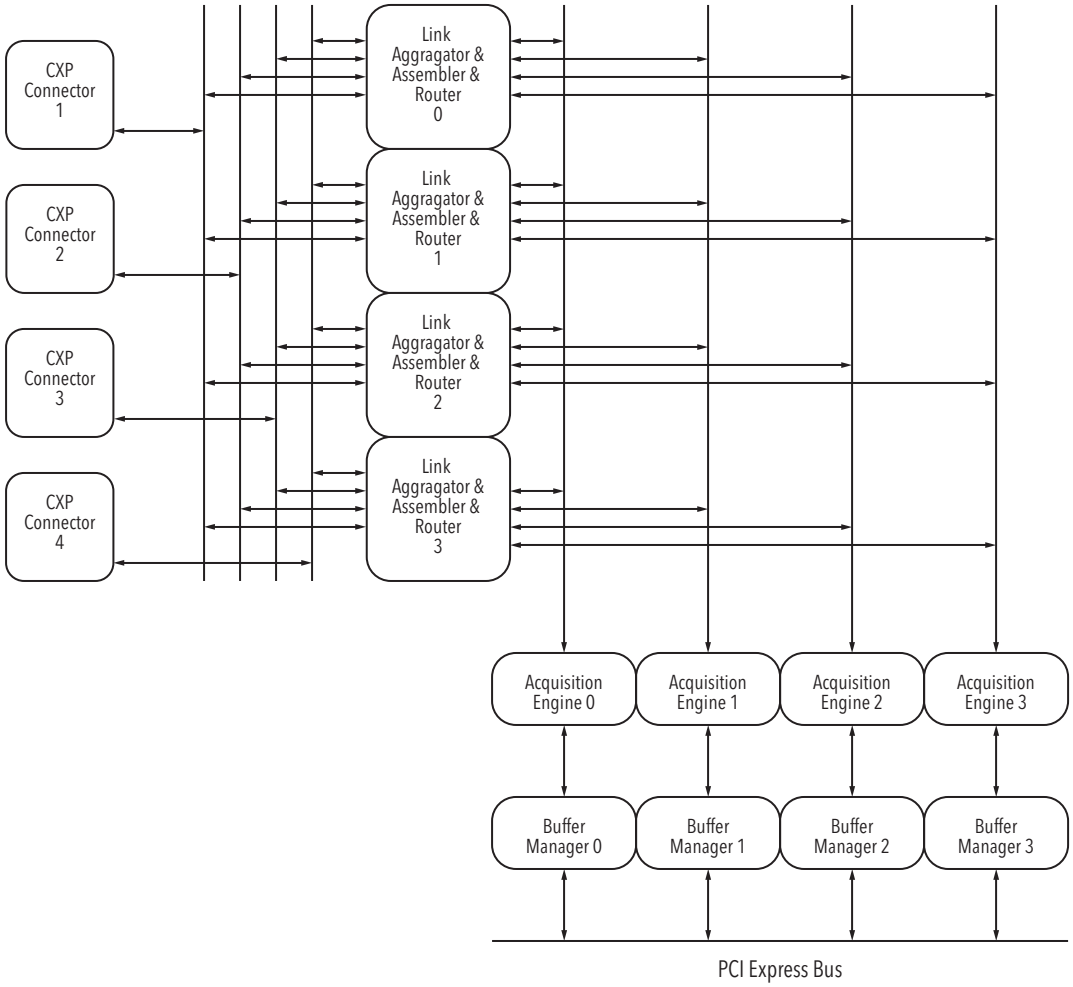

Figure 1-5 Routing Between CXP Links and VFGs

Figure 1-5 may look confusing, but all of the routing details are handle automatically by software, there is no need to worry about programming the routing tables manually in your software. However, it is helpful to understand the underlying structure of the Cyton.

Note: In certain situations, a VFG may have no CXP links routed to it, as they may all be used by other VFGs. In this case, the VFG with no links can not be used for acquisition.

1.6 The BitFlow CXP Models

There are two models of the Cyton-CXP. Table 1-1 illustrates the capabilities of each model.

Table 1-1 BitFlow CXP Models

Capability	AON-PC2-CXP1	CLX-PC2-CXP1	CLX-PC3-CXP2	CLX-PC3-CXP4	CLX-PC3-FXP4	CYT-PC2-CXP2	CYT-PC2-CXP4
Number of 3.125 Gb/S links	1	1	2	4	4*	2	4
Number of 6.25 Gb/S links	1	1	2	4	4*	2	4
Number of 12.5 Gb/S links	0	1	2	4	4**	0	0
Number of single link cameras	1	1	2	4	4	2	4
Number of dual link cameras	0	0	1	2	2	1	2
Number of quad link cameras	0	0	0	1	1	0	1
Number of Virtual Frame Grabbers	1	1	2	4	4	2	4
Number of independent trigger inputs	1	1	2	4	4	2	4
Number of independent encoder inputs	1	1	2	4	4	2	4
Number of PCI configurations (devices)	1	1	2	4	4	2	4
Maximum DMA bandwidth	1.5 GB/S	1.75 GB/S	7.2 GB/S	7.2 GB/S	7.2 GB/S	3.5 GB/S	3.5 GB/S

* CoaXPress over Fiber always uses a 10 Gb/S data rate. Cameras running CXP-3, CXP-6 or other speed less than CXP-12 will actually a data over the fiber at 10 Gb/S. For example, CXP-3 packets will be moving over the fiber at 10 Gb/S, there will be more dead time between packets, so the average data rate will data will be the equivalent of CXP-3 over copper.

** CoaXPress over Fiber uses 10 Gb/S fiber. However, it can still support CXP-12 data rates. The reason is that CXP over copper uses 8B/10B encoding, and CXP over fiber uses 64B/66B encoding, which more efficient.

The StreamSync Acquisition Engine

Chapter 2

2.1 Introduction

The StreamSync system consists of an Acquisition Engine and a Buffer Manager. The StreamSync system was first released on the Cyton-CXP and is a departure from previous BitFlow frame grabbers. Subsequently it was ported to the Axion, and will be used on all new frame grabbers moving forward. The StreamSync system is a start-from-scratch complete redesign of the acquisition and DMA parts of a frame grabber. BitFlow used its years of experience in this area to design a next generation, super efficient capture system.

Currently the StreamSync engine is used on the Aon-CXP, the Axion-CL, Claxon-CXP, Claxon-FXP, and the Cyton-CXP.

From a software perspective, the StreamSync system is compatible with the previous BitFlow products. However, digging deeper, these new system have a lot more power and flexibility. These new features will be described in the following sections.

The StreamSync system has many improvements over previous systems. The main improvements are:

- Efficient support for variable sized images with fast context switches between frames
- Per frame control of acquisition properties (AOI specifically)
- Hardware control of image sequencing
- Enhanced debug capabilities
- Efficient support for on-demand buffer allocation (GenICam model)
- Gracefully recovery from dropped packets (either on the input side or the DMA side)

This chapter describes the StreamSync Acquisition Engine while the next chapter describes the StreamSync Buffer Manager.

2.2 The StreamSync Acquisition Engine World

We are used to the concept that images have an X and a Y dimension. The Acquisition Engine expands on this concept by adding two further dimension Z and V. The Z dimension controls a sequence of frames or "Volume" of frames. The V dimension controls a sequence of volumes, or "Hypervolume". Figure 2-1 illustrates these concepts.

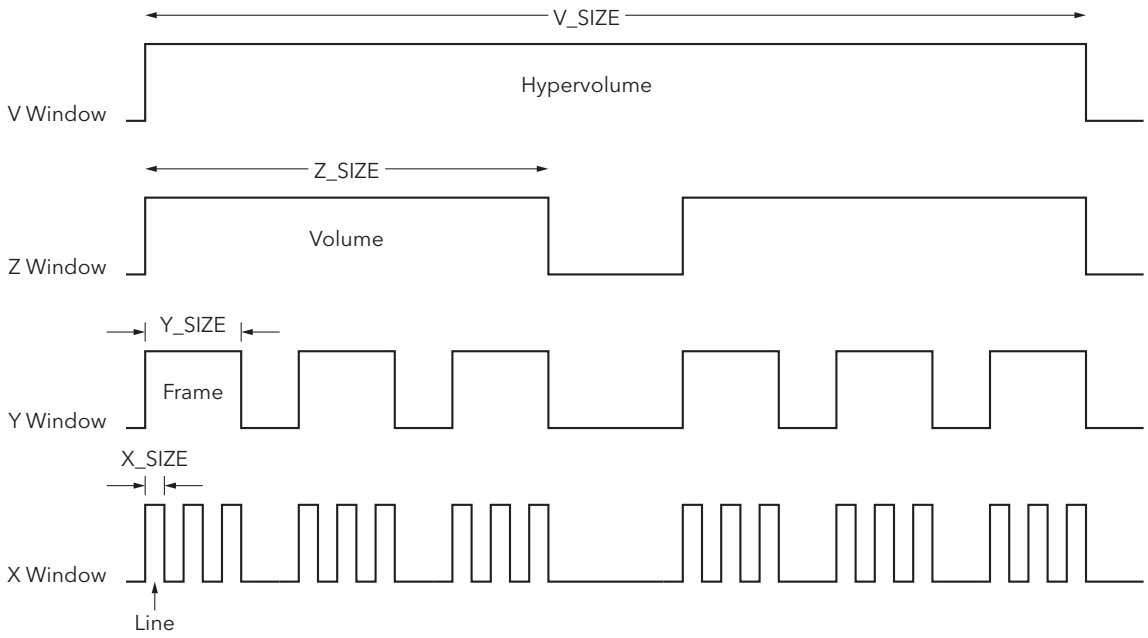

Figure 2-1 StreamSync Acquisition Engine Dimensions

The size of the X window, i.e. the number of pixels per line, is controlled by the X_SIZE register. The size of the Y window, i.e. number of lines per frame, is controlled by the Y_SIZE register. The size of the Z window, i.e. number of frames per volume, is controlled by the Z_SIZE register. Finally, the size of the V window, i.e. number of volumes to acquire, is controlled by the V_SIZE register. Note that the size of the Y window and the Z window can be dynamically controlled by external triggers, see below for more details.

2.2.1 Controlling the StreamSync Acquisition Engine

Acquisition of images is controlled by the AE_RUN_LEVEL register. The run level controls the conditions under which the Acquisition Engine will start or stop acquiring image data. Acquisition can be idle, which means nothing will be acquired, or it can be running, which means data will be acquired when the engine is inside the V, Z,Y and X windows. There are various conditions which control whether the engine is inside or outside of these windows.

Acquisition can be aborted on any X, Y, Z, or V boundary. The choice of boundary depends on whether one wants to abort immediate, which can cause acquisition of incomplete frames, or one wants to stop at the end of the line/frame/volume, which provides a more graceful end to acquisition.

It's easiest to think of the Acquisition Engine level in terms of a state machine. When a window is "opened" the run level moves down to the next state below. When a window "closes", the run level moves to the state above. If the window at the top level closes, the run level goes to idle. Figure 2-2 illustrates this type of state machine:

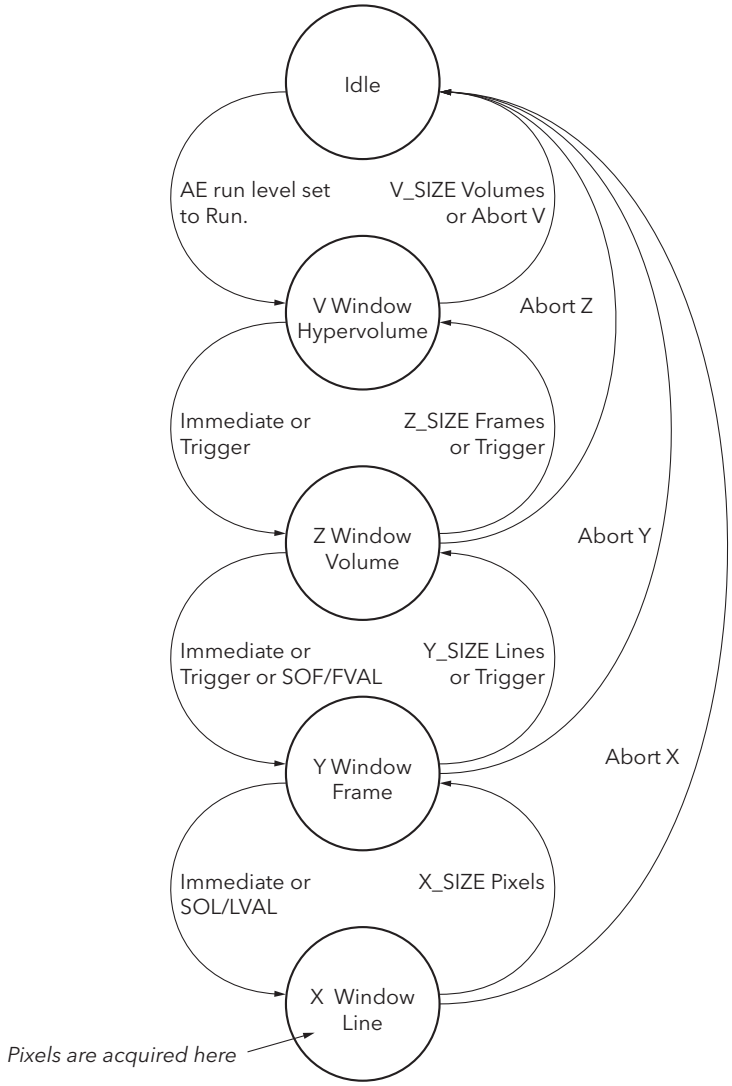

Figure 2-2 Acquisition Engine Run Level

The action that causes a window to be opened or closed depends on the type of window. Some windows can be opened in more than one way. For example the Y window can be opened when a Start Of Frame (SOF) packet (or FVAL from a CL camera) is sent from the camera, or it can be opened by a trigger (all SOF packets are ignored

until the trigger condition is met) or it can just be opened immediately, as soon the Acquisition Engine level is inside the X window (i.e. the stat above). Table 2-1 enumerates all of these conditions..

Table 2-1 Open Close Conditions

Window	Open	Close
V	AE run level set to Run	Abort V, V_SIZE volumes
Z	Trigger, Immediate	Trigger, Abort Z, Z_SIZE frame
Y	Trigger, SOF/FVAL, Immediate	Trigger, Abort Y, Y_SIZE lines
X	SOL/LVAL, Immediate	Abort X, X_SIZE pixels

2.2.2 Observing the StreamSync Acquisition Engine

The state of the Acquisition Engine can be observed at any time. The register AE_LEVEL indicates the current run level of the Acquisition Engine. In other words, this register returns the current state as shown in Figure 2-2. While this is not very useful in a free-running situation, as the value will be changing constantly, it can be very helpful debugging if the system gets stuck (e.g. waiting for a trigger).

2.2.3 Synchronizing the StreamSync Acquisition Engine With a CXP Camera

Normally acquisition is synchronized with camera by special CXP header packets called Start Of Frame (SOF) and Start Of Line (SOL). The Acquisition Engine will synchronize its Y window (frame) with the SOF and X window with the SOL. This means that all packets from the camera will be dropped until the SOF is seen (causing the Acquisition Engine to open the Y window), and then packets are further dropped until SOL must be seen (opening the X window). Each line requires an SOL packet. This process keeps the Acquisition Engine synchronized to the camera even if packets are dropped. This functionality can be enable/disable by the ZSYNC and YSYNC bits.

2.2.4 Synchronizing the StreamSync Acquisition Engine With a Camera Link Camera

Normally acquisition is synchronized with camera by FVAL (Frame VALid) and the LVAL (Line VALid). The Acquisition Engine will synchronize its Y window (frame) with the FVAL and X window (line) with the LVAL. This means that all pixels from the camera will be dropped until the FVAL is seen (causing the Acquisition Engine to open the Y window), and then packets are further dropped until LVAL must be seen (opening the X window). Each line requires an LVAL edge. This process keeps the Acquisition Engine synchronized to the camera even if packets are dropped. This functionality can be enable/disable by the ZSYNC and YSYNC bits.

Note: For line scan camera, FVAL is not used and the Y Window will open either immediately in free-run mode, or based on a trigger eddge.

2.2.5 Regions Of Interest (ROI) with the StreamSync Acquisition Engine.

The Acquisition Engine support capturing a subwindow or ROI of the image that the camera is putting out. The Y_SIZE and X_SIZE registers control how many lines and pixels are acquired per frame, regardless of the actual frames size coming out of the camera. Further there are Y_OFFS and X_OFFS registers which can locate the subwindow anywhere inside of the camera's frame. These concepts are show in Figure 2-3.

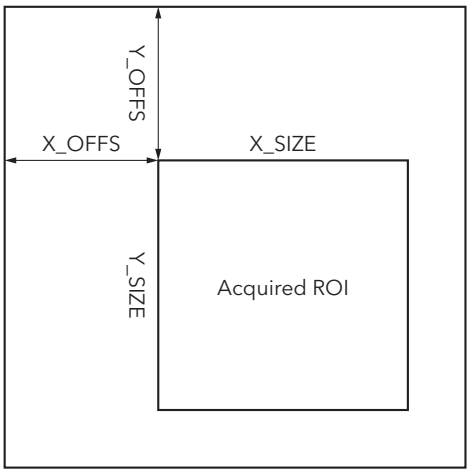

Figure 2-3 Acquisition Engine ROI

Similarly there is a Z_OFFS register which if non-zero can cause the board to discard a certain number of frames before starting an acquisition of a sequence. This concept is illustrated in Figure 2-4.

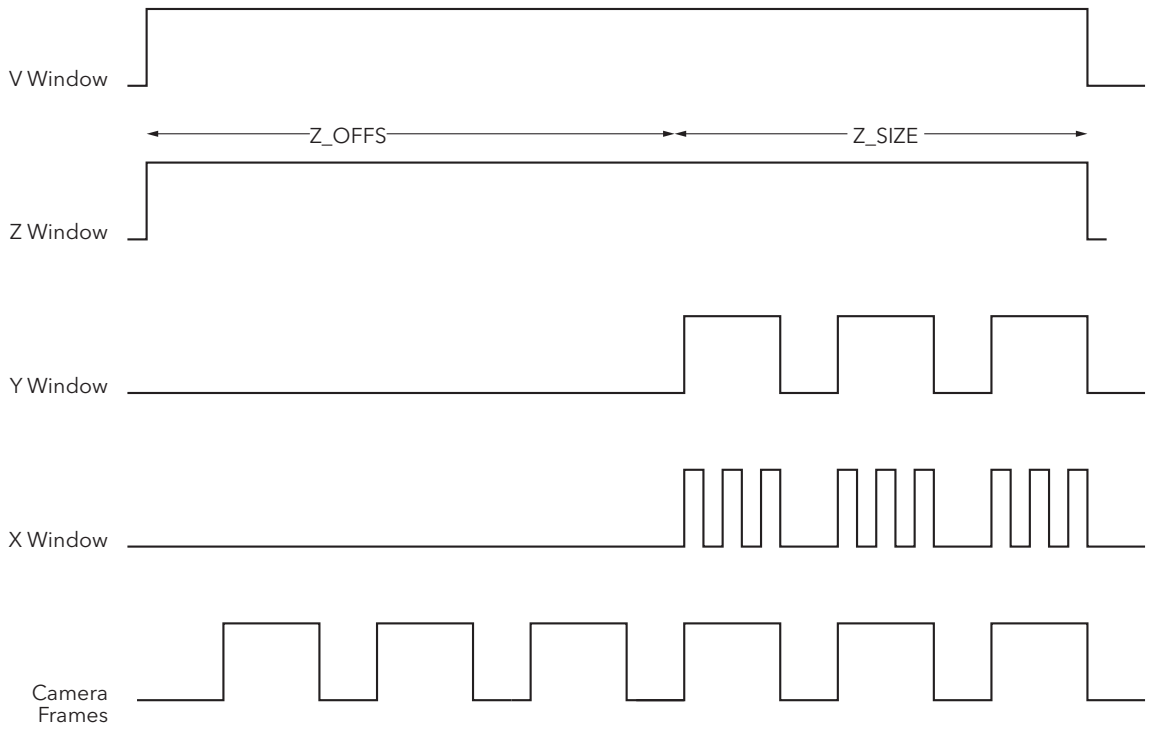

Figure 2-4 Z_OFFS Illustration

2.3 Window Interrupts

Interrupts at the start and end of each major window are available for use by host processes. The X, Y and Z windows (line, frame volume) all can provide interrupts. There are no interrupts for the V Window (hypervolume). Figure 2-5 Show the relationship between the interrupts and the acquisition Windows.

Note: The labels in italics in Figure 2-5 are are the actual interrupt names that can be used with the BitFlow SDK function calls.

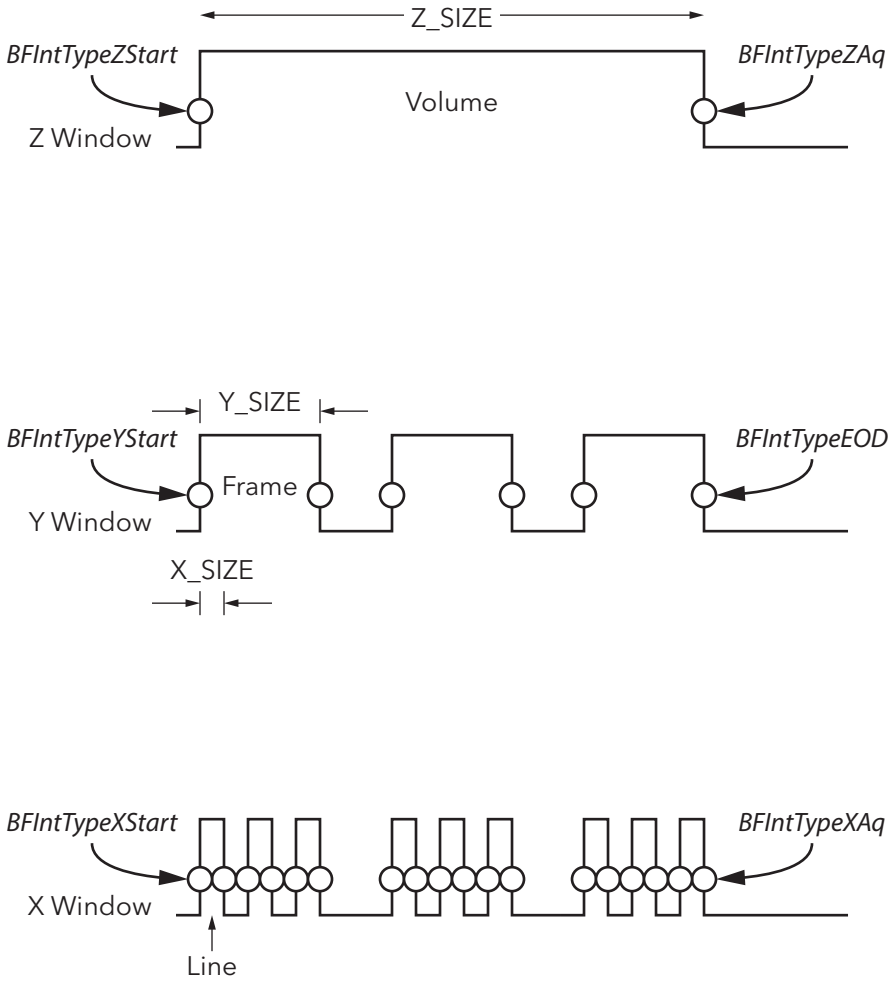

Figure 2-5 X/Y/Z Interrupts

2.4 AE State Machine

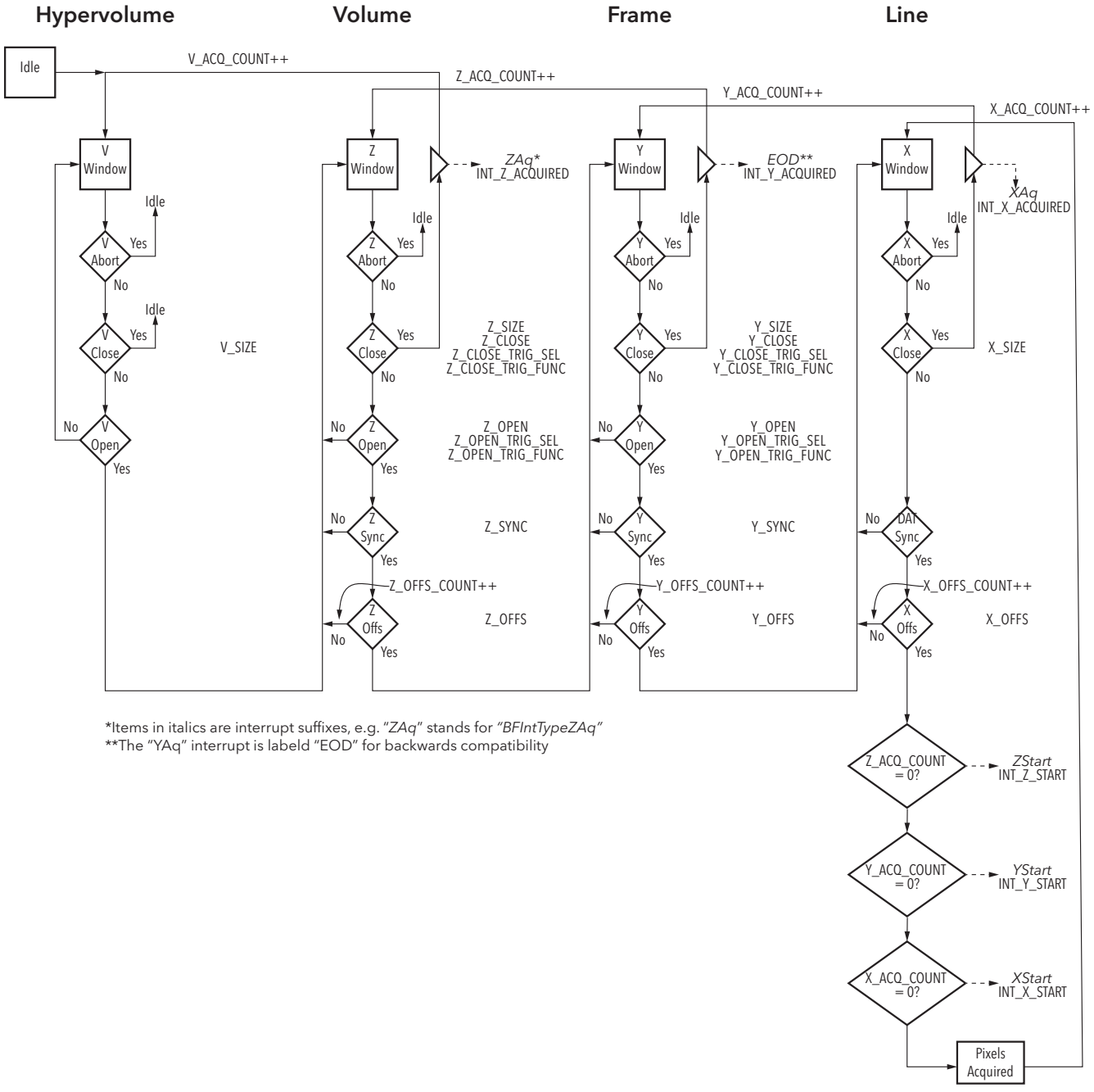

Figure 2-6 AE State Machine

2.5 Triggering the StreamSync Acquisition Engine

One of the areas where the power of the Acquisition Engine is really seen is with regards to triggering. There are many more ways to use triggers in the Acquisition Engine. Primarily triggers can be used to “open” a window and/or to “close” a window. For example, a trigger could be used to start the acquisition of each frame and/or end the acquisition of each frame.

Further, a trigger could be use to start the acquisition of each volume (sequence of frames) and/or end the acquisition of a volume. Further flexibility comes from the fact that the source for each event (i.e. open or close) can be different or the same. This means a frame could be started with one trigger or ended with another, or the frame could start on the rising edge and end on the falling edge of the same trigger. Please refer to Figure 2-2 for more information on how a trigger can be used to change the state of the Acquisition Engine.

2.6 Comparing the StreamSync Acquisition Engine to Other BitFlow products

While the Acquisition Engine might seem very complex, it is actually quite simple to use and has considerably more power than previous acquisition engines used on all previous BitFlow frame grabbers. From a software point of view, the BitFlow API hides the differences between the traditional acquisition systems and the newer Acquisition Engine. However, for users that desire more flexibility and are willing to do some lower level coding, the Acquisition Engine can handle almost any acquisition scenario.

For users who were already doing some lower level programming using other BitFlow products, it's helpful to see how this new system relates to the tradition acquisition engine.Table 2-2 shows some examples of the traditional and the new system.

Table 2-2 Comparing Traditional and New Acquisition Systems

Traditional Command	X_SIZE	Y_SIZE	Z_SIZE	V_SIZE	AE_RUN_ LEVEL
Snap	Camera Width	Camera Height	1	1	Run
Grab	Camera Width	Camera Height	1	0xffffff	Run
Grab N frames	Camera Width	Camera Height	N	1	Run
Freeze					Abort Z
Abort					Abort X

2.7 AE_CON

Bit	Name
0	AE_RUN_LEVEL
1	AE_RUN_LEVEL
2	AE_RUN_LEVEL
3	AE_RUN_LEVEL
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	CLR_ACQ_COUNT
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

AE_RUN_LEVEL R/W, AE_CON[3..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This is the main control for starting/aborting acquisition. Writing this register changes the current run level. Reading this register returns the current run level command (not the current status). The abort run levels exit acquisition on a clean boundary. V exits on a volume boundary, Z on a frame boundary, Y on a line boundary, X on a 128-byte data boundary.

AE_RUN_LEVEL	Meaning
0 (0000b)	System is idle
1 (0001b)	Run - start running (i.e. acquiring)
2 (0010b)	Abort V - stop at the end of the next volume
3 (0011b)	Abort Z - stop at the end of the next frame
4 (0100b)	Abort Y - stop at the end of the next line
5 (0101b)	Abort X - stop at the end of the next 128-byte block

CLR_ACQ_COUNT RO, AE_CON[9], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Write a 1 to this bitfield will reset to 0 the following registers V_ACQ_COUNT, Z_ACQ_COUNT, Y_ACQ_COUNT and X_ACQ_COUNT.

2.8 AE_STATUS

Bit	Name
0	AE_STATE
1	AE_STATE
2	AE_STATE
3	Reserved
4	AE_FIFO_OVERFLOW
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

AE_STATE RO, AE_STATUS[2..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register indicates the current run level of the acquisition engine. The following table shows the meanings of each state.

AE_STATE	Meaning
0 (000b)	Idle - System is idle
1 (001b)	System is inside the V window
2 (010b)	System is inside the Z window
3 (011b)	System is inside the Y window
4 (100b)	System is inside the X window

AE_FIFO_ OVERFLOW RO, AE_STATUS[4], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

If this bit is 1, the FIFO between acquisition engine and packet generation overflowed. The acquisition engine will abort.

2.9 AE_STREAM_SEL

Bit	Name
0	STREAM_SEL
1	STREAM_SEL
2	STREAM_SEL
3	STREAM_SEL
4	STREAM_SEL
5	STREAM_SEL
6	STREAM_SEL
7	STREAM_SEL
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	USE_SYNTHETIC_FRAME

STREAM_SEL R/W, AE_STREAM_SEL[7..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Program this register to the stream aggregator that this acquisition engine should get its data from. Currently only the values 0 to 3 are supported. Generally this register should be programmed to correspond to the VFG number that is being used to access the acquisition engine. For example, for VFG1 set this register to 1.

USE_ SYNTHETIC_ FRAME R/W, AE_STREAM_SEL[31], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Use the Synthetic Frame generator instead of the camera.

2.10 V_WIN_DIM

Bit	Name
0	V_SIZE
1	V_SIZE
2	V_SIZE
3	V_SIZE
4	V_SIZE
5	V_SIZE
6	V_SIZE
7	V_SIZE
8	V_SIZE
9	V_SIZE
10	V_SIZE
11	V_SIZE
12	V_SIZE
13	V_SIZE
14	V_SIZE
15	V_SIZE
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

V_SIZE R/W, V_WIN_DIM[15..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register defines size of the V window, that is, the number of volumes to acquire. A value of 0XFFFF means infinite. When set to infinite, the acquisition engine can be stopped by writing AE_RUN_LEVEL.

The most common setting for this field is either 1 or 0xFFFF.

This register is writable only when AE_STATE is 0 (idle). Writes to this field will be ignored if AE_STATE is not 0.

2.11 Z_WIN_CON

Bit	Name
0	Z_CLOSE_TRIG_FUNC
1	Z_CLOSE_TRIG_FUNC
2	Z_CLOSE_TRIG_FUNC
3	Z_CLOSE_TRIG_FUNC
4	Z_CLOSE_TRIG_SEL
5	Z_CLOSE_TRIG_SEL
6	Z_CLOSE_TRIG_SEL
7	Z_CLOSE_TRIG_SEL
8	Z_CLOSE
9	Z_CLOSE
10	Z_CLOSE
11	Z_CLOSE
12	Z_OPEN_TRIG_FUNC
13	Z_OPEN_TRIG_FUNC
14	Z_OPEN_TRIG_FUNC
15	Z_OPEN_TRIG_FUNC
16	Z_OPEN_TRIG_SEL
17	Z_OPEN_TRIG_SEL
18	Z_OPEN_TRIG_SEL
19	Z_OPEN_TRIG_SEL
20	Z_OPEN
21	Z_OPEN
22	Z_OPEN
23	Z_OPEN
24	Z_SYNC
25	Z_SYNC
26	Z_SYNC
27	Z_SYNC
28	Reserved
29	Reserved
30	Reserved
31	Reserved

Z_CLOSE_TRIG_FUNC R/W, Z_WIN_CON[3..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register determines which trigger change (if any) will end the Z window.

Z_CLOSE_TRIG_FUNC	Meaning
0 (0000b)	Rising edge of trigger
1 (0001b)	Falling edge of trigger
2 (0010b)	Trigger is high
3 (0011b)	Trigger is low
4 (0100b)	Either edge of trigger

Z_CLOSE_TRIG_SEL R/W, Z_WIN_CON[7..4], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Selects which trigger will control the end the Z window.

Z_CLOSE_TRIG_SEL	Meaning
0 (0000b)	The selected trigger (VFGx_TRIG_SEL)
1 (0001b)	The selected encoder A (VFGx_ENCA_SEL)
2 (0010b)	The selected encoder B (VFGx_ENCB_SEL)
3 (0011b)	The selected encoder divider (VFGx_ENCDIV_SEL)
4 (0100b)	The selected encoder quad (VFGx_ENCQ_SEL)

Z_CLOSE R/W, Z_WIN_CON[11..8], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This field specifies how the Z window closes. Possible values are: 0 - size mode, 1 - trigger mode.

If size mode is specified, the acquisition engine waits for Z_SIZE number of complete frames, it then closes the Z window and then checks to see if t are more volumes to acquire.

If trigger mode is specified, the trigger is selected by Z_CLOSE_TRIG_SEL and the conditioning function is specified by Z_CLOSE_TRIG_FUNC. The acquisition engine waits for the trigger condition to be satisfied, then continues acquiring to the next frame boundary, it then closes the Z window and then checks to see if t are more volumes to acquire.

Z_OPEN_TRIG_FUNC R/W, Z_WIN_CON[15..12], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register determines which trigger change (if any) will start Z window.

Z_OPEN_TRIG_FUNC	Meaning
0 (0000b)	Rising edge of trigger
1 (0001b)	Falling edge of trigger
2 (0010b)	Trigger is high
3 (0011b)	Trigger is low
4 (0100b)	Either edge of trigger

Z_OPEN_TRIG_SEL R/W, Z_WIN_CON[19..16], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Selects which trigger will control the start Z window.

Z_OPEN_TRIG_SEL	Meaning
0 (0000b)	The selected trigger (VFGx_TRIG_SEL)
1 (0001b)	The selected encoder A (VFGx_ENCA_SEL)
2 (0010b)	The selected encoder B (VFGx_ENCB_SEL)
3 (0011b)	The selected encoder divider (VFGx_ENCDIV_SEL)
4 (0100b)	The selected encoder quad (VFGx_ENCQ_SEL)

Z_OPEN R/W, Z_WIN_CON[23..20], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This field specifies how the Z window starts. Possible values are: 0 - immediate mode, 1 - trigger mode.

If immediate mode is specified, no trigger synchronization is required. The acquisition engine waits for any frame sync requirements, opens the Z window, then starts the setup of the Y window.

If trigger mode is specified, the trigger is selected by Z_OPEN_TRIG_SEL and the conditioning function is specified by Z_OPEN_TRIG_FUNC. The acquisition engine waits for the trigger condition to be satisfied, opens the Z window, then starts the setup of the Y window.

Z_SYNC R/W, Z_WIN_CON[27..24], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This field enforces the data-synchronization of streaming video to the acquisition engine for each individual frame in the z window. The following table shows explains this field.

Z_SYNC	Meaning
0	No synchronization. All streaming packets received after the Z Window are open will be acquired as part of the current frame.
1	Start Of Frame (SOF) synchronization. All pixels received before the SOF will be ignored. This conditions is enforced for each frame in the Z window. On the Axion, SOF is generated by the FVAL signal coming from the camera.

2.12 Z_WIN_DIM

Bit	Name
0	Z_SIZE
1	Z_SIZE
2	Z_SIZE
3	Z_SIZE
4	Z_SIZE
5	Z_SIZE
6	Z_SIZE
7	Z_SIZE
8	Z_SIZE
9	Z_SIZE
10	Z_SIZE
11	Z_SIZE
12	Z_SIZE
13	Z_SIZE
14	Z_SIZE
15	Z_SIZE
16	Z_OFFS
17	Z_OFFS
18	Z_OFFS
19	Z_OFFS
20	Z_OFFS
21	Z_OFFS
22	Z_OFFS
23	Z_OFFS
24	Z_OFFS
25	Z_OFFS
26	Z_OFFS
27	Z_OFFS
28	Z_OFFS
29	Z_OFFS
30	Z_OFFS
31	Z_OFFS

Z_SIZE R/W, Z_WIN_DIM[15..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of frames (Y windows) to acquire per sequence (Z windows). The acquisition of frames will only start after Z_OFFS frames have been skipped after the Z window is opened. Note this register is actually 24 bits, the upper 8 MSB are located in the Z_SIZE_MSB register.

Z_OFFS R/W, Z_WIN_DIM[31..16], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The number of frames (Y windows) to skip before starting acquisition after the Z window has been opened. Note this register is actually 24 bits, the upper 8 MSB are located in the Z_OFFS_MSB register.

2.13 Z_WIN_DIM_EXT

Bit	Name
0	Z_SIZE_MSB
1	Z_SIZE_MSB
2	Z_SIZE_MSB
3	Z_SIZE_MSB
4	Z_SIZE_MSB
5	Z_SIZE_MSB
6	Z_SIZE_MSB
7	Z_SIZE_MSB
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Z_OFFS_MSB
17	Z_OFFS_MSB
18	Z_OFFS_MSB
19	Z_OFFS_MSB
20	Z_OFFS_MSB
21	Z_OFFS_MSB
22	Z_OFFS_MSB
23	Z_OFFS_MSB
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

Z_SIZE_MSB R/W, Z_WIN_DIM_EXT[7..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Extends the Z_SIZE register by 8 more bits.

Z_OFFS_MSB R/W, Z_WIN_DIM_EXT[23..16], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Extends the Z_OFFS register by 8 more bits.

2.14 Y_INT_DEC

Bit	Name
0	Y_INT_DEC_COUNT
1	Y_INT_DEC_COUNT
2	Y_INT_DEC_COUNT
3	Y_INT_DEC_COUNT
4	Y_INT_DEC_COUNT
5	Y_INT_DEC_COUNT
6	Y_INT_DEC_COUNT
7	Y_INT_DEC_COUNT
8	Y_INT_DEC_COUNT
9	Y_INT_DEC_COUNT
10	Y_INT_DEC_COUNT
11	Y_INT_DEC_COUNT
12	Y_INT_DEC_COUNT
13	Y_INT_DEC_COUNT
14	Y_INT_DEC_COUNT
15	Y_INT_DEC_COUNT
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Y_INT_DEC_RST
29	Y_INT_DEC_RST
30	Y_INT_DEC_MODE
31	Y_INT_DEC_MODE

Y_INT_DEC_COUNT R/W, Y_INT_DEC[15..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

When Y interrupt decimate mode is enable, the register determines the decimation amount. In other words, Y_INT_DEC_COUNT interrupts must occur before the board emits a real interrupt.

Y_INT_DEC_RST R/W, Y_INT_DEC[29..28], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Resets the Y interrupt decimation counter to 0.

Y_INT_DEC_MODE R/W, Y_INT_DEC[29..28], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register controls how Y interrupts are decimated.

Y_DEC_MODE	Meaning
0 (0000b)	Y interrupts are not decimated
1 (0001b)	Y interrupts are decimated by Y_INT_DEC_COUNT
2 (0010b)	reserved
3 (0011b)	reserved

2.15 Y_WIN_CON

Bit	Name
0	Y_CLOSE_TRIG_FUNC
1	Y_CLOSE_TRIG_FUNC
2	Y_CLOSE_TRIG_FUNC
3	Y_CLOSE_TRIG_FUNC
4	Y_CLOSE_TRIG_SEL
5	Y_CLOSE_TRIG_SEL
6	Y_CLOSE_TRIG_SEL
7	Y_CLOSE_TRIG_SEL
8	Y_CLOSE
9	Y_CLOSE
10	Y_CLOSE
11	Y_CLOSE
12	Y_OPEN_TRIG_FUNC
13	Y_OPEN_TRIG_FUNC
14	Y_OPEN_TRIG_FUNC
15	Y_OPEN_TRIG_FUNC
16	Y_OPEN_TRIG_SEL
17	Y_OPEN_TRIG_SEL
18	Y_OPEN_TRIG_SEL
19	Y_OPEN_TRIG_SEL
20	Y_OPEN
21	Y_OPEN
22	Y_OPEN
23	Y_OPEN
24	Y_SYNC
25	Y_SYNC
26	Y_SYNC
27	Y_SYNC
28	Reserved
29	Reserved
30	Reserved
31	Reserved

Y_CLOSE_TRIG_FUNC R/W, Y_WIN_CON[3..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register determines which trigger change (if any) will end the Y window.

Y_CLOSE_TRIG_FUNC	Meaning
0 (0000b)	Rising edge of trigger
1 (0001b)	Falling edge of trigger
2 (0010b)	Trigger is high
3 (0011b)	Trigger is low
4 (0100b)	Either edge of trigger

Y_CLOSE_TRIG_SEL R/W, Y_WIN_CON[7..4], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Selects which trigger will control the end the Y window.

Y_CLOSE_TRIG_SEL	Meaning
0 (0000b)	The selected trigger (VFGx_TRIG_SEL)
1 (0001b)	The selected encoder A (VFGx_ENCA_SEL)
2 (0010b)	The selected encoder B (VFGx_ENCB_SEL)
3 (0011b)	The selected encoder divider (VFGx_ENCDIV_SEL)
4 (0100b)	The selected encoder quad (VFGx_ENCQ_SEL)
5 (0101b)	The falling edge of FVAL will close the window (Axion only)

Y_CLOSE R/W, Y_WIN_CON[11..8], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This field specifies how the Y window closes. Possible values are: 0 - size mode, 1 - trigger mode, 2 - size or trigger mode.

If Y_CLOSE = 0, the acquisition engine waits for Y_SIZE_SIZE number of complete lines, it then closes the Y window and then checks to see if there are more frames to acquire.

If Y_CLOSE = 1, the trigger is selected by Y_CLOSE_TRIG_SEL and the conditioning function is specified by Y_CLOSE_TRIG_FUNC. The acquisition engine waits for the trigger condition to be satisfied, then continues acquiring to the next line boundary, it then closes the Y window and then checks to see if there are more frames to acquire.

If Y_CLOSE = 2, either condition above will close the window. Whichever conditions comes first will close the window (end the frame). This mode is used to acquire variable sized images where the trigger controls the frame size. Usually the rising edge of trigger opens the window (starts the frame) lines are acquired until either the trigger goes low, or Y_SIZE lines have been acquired (i.e. the maximum frame size has been reached).

Y_OPEN_TRIG_FUNC R/W, Y_WIN_CON[15..12], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register determines which trigger change (if any) will start Y window.

Y_OPEN_TRIG_FUNC	Meaning
0 (0000b)	Rising edge of trigger
1 (0001b)	Falling edge of trigger
2 (0010b)	Trigger is high
3 (0011b)	Trigger is low
4 (0100b)	Either edge of trigger

Y_OPEN_TRIG_SEL R/W, Y_WIN_CON[19..16], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Selects which trigger will control the start Y window.

Y_OPEN_TRIG_SEL	Meaning
0 (0000b)	The selected trigger (VFGx_TRIG_SEL)
1 (0001b)	The selected encoder A (VFGx_ENCA_SEL)
2 (0010b)	The selected encoder B (VFGx_ENCB_SEL)
3 (0011b)	The selected encoder divider (VFGx_ENCDIV_SEL)
4 (0100b)	The selected encoder quad (VFGx_ENCQ_SEL)

Y_OPEN R/W, Y_WIN_CON[23..20], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This field specifies how the Y window starts. Possible values are: 0 - immediate mode, 1 - trigger mode.

If immediate mode is specified, no trigger synchronization is required. The acquisition engine waits for any line sync requirements, opens the Y window, then starts the setup of the X window.

If trigger mode is specified, the trigger is selected by Y_OPEN_TRIG_SEL and the conditioning function is specified by Y_OPEN_TRIG_FUNC. The acquisition engine waits for the trigger condition to be satisfied, opens the Y window, then starts the setup of the X window.

Y_SYNC R/W, Y_WIN_CON[27..24], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This field enforces the data-synchronization of streaming video to the acquisition engine for each individual line in the y window. The following table explains this field.

Y_SYNC	Meaning
0	No synchronization. All streaming packets received after the Y Window are open will be acquired as part of the current line.
1	Start Of Line (SOL) synchronization. All pixels received before the SOL will be ignored. This conditions is enforced for each line in the Y window. On the Axion, SOL is generated from the LVAL signal.

2.16 Y_WIN_DIM

Bit	Name
0	Y_SIZE
1	Y_SIZE
2	Y_SIZE
3	Y_SIZE
4	Y_SIZE
5	Y_SIZE
6	Y_SIZE
7	Y_SIZE
8	Y_SIZE
9	Y_SIZE
10	Y_SIZE
11	Y_SIZE
12	Y_SIZE
13	Y_SIZE
14	Y_SIZE
15	Y_SIZE
16	Y_OFFS
17	Y_OFFS
18	Y_OFFS
19	Y_OFFS
20	Y_OFFS
21	Y_OFFS
22	Y_OFFS
23	Y_OFFS
24	Y_OFFS
25	Y_OFFS
26	Y_OFFS
27	Y_OFFS
28	Y_OFFS
29	Y_OFFS
30	Y_OFFS
31	Y_OFFS

Y_SIZE R/W, Y_WIN_DIM[15..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of lines per frame (Y window) to acquire. This number is only acquired after the Y window is opened and after Y_OFFS lines have been skipped.

Y_OFFS R/W, Y_WIN_DIM[31..16], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of lines to skip before starting the acquisition of lines (after the Y windows is opened).

2.17 Y_WIN_DIM_EXT

Bit	Name
0	Y_SIZE_MSB
1	Y_SIZE_MSB
2	Y_SIZE_MSB
3	Y_SIZE_MSB
4	Y_SIZE_MSB
5	Y_SIZE_MSB
6	Y_SIZE_MSB
7	Y_SIZE_MSB
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Y_OFFS_MSB
17	Y_OFFS_MSB
18	Y_OFFS_MSB
19	Y_OFFS_MSB
20	Y_OFFS_MSB
21	Y_OFFS_MSB
22	Y_OFFS_MSB
23	Y_OFFS_MSB
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

Y_SIZE_MSB R/W, Y_WIN_DIM_EXT[7..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Extends the Y_SIZE register by 8 more bits.

Y_OFFS_MSB R/W,Y_WIN_DIM_EXT[23..16], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Extends the Y_OFFS register by 8 more bits.

2.18 X_WIN_DIM

Bit	Name
0	X_SIZE
1	X_SIZE
2	X_SIZE
3	X_SIZE
4	X_SIZE
5	X_SIZE
6	X_SIZE
7	X_SIZE
8	X_SIZE
9	X_SIZE
10	X_SIZE
11	X_SIZE
12	X_SIZE
13	X_SIZE
14	X_SIZE
15	X_SIZE
16	X_OFFS
17	X_OFFS
18	X_OFFS
19	X_OFFS
20	X_OFFS
21	X_OFFS
22	X_OFFS
23	X_OFFS
24	X_OFFS
25	X_OFFS
26	X_OFFS
27	X_OFFS
28	X_OFFS
29	X_OFFS
30	X_OFFS
31	X_OFFS

X_SIZE R/W, X_WIN_DIM[15..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of 16-byte data words to acquired per line (X window). This number is only acquired after the X window is opened and after X_OFFS words have been skipped.

X_OFFS R/W, X_WIN_DIM[31..16], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of 16-byte data words to skip per line (after the X window is opened).

2.19 X_WIN_DIM_EXT

Bit	Name
0	X_SIZE_MSB
1	X_SIZE_MSB
2	X_SIZE_MSB
3	X_SIZE_MSB
4	X_SIZE_MSB
5	X_SIZE_MSB
6	X_SIZE_MSB
7	X_SIZE_MSB
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	X_OFFS_MSB
17	X_OFFS_MSB
18	X_OFFS_MSB
19	X_OFFS_MSB
20	X_OFFS_MSB
21	X_OFFS_MSB
22	X_OFFS_MSB
23	X_OFFS_MSB
24	X_SHIFT
25	X_SHIFT
26	X_SHIFT
27	X_SHIFT
28	X_SHIFT
29	X_SHIFT_DIR
30	Reserved
31	Reserved

X_SIZE_MSB R/W, X_WIN_DIM_EXT[7..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Extends the X_SIZE register by 8 more bits.

X_OFFS_MSB R/W,X_WIN_DIM_EXT[23..16], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Extends the X_OFFS register by 8 more bits.

X_SHIFT R/W,X_WIN_DIM_EXT[24..18], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register can be program to shift valid an image line left or right with respect to the start of LVAL (Camera Link) or SOL (CoaXPress). The units are pixels. To shift more than 2^5, us the X_OFFSET register. Use X_SHIFT_DIR to control the direction.

X_SHIFT_DIR R/W,X_WIN_DIM_EXT[29], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the direction of the line shift set by the X_SHIFT register.

2.20 V_ACQUIRED

Bit	Name
0	V_ACQ_COUNT
1	V_ACQ_COUNT
2	V_ACQ_COUNT
3	V_ACQ_COUNT
4	V_ACQ_COUNT
5	V_ACQ_COUNT
6	V_ACQ_COUNT
7	V_ACQ_COUNT
8	V_ACQ_COUNT
9	V_ACQ_COUNT
10	V_ACQ_COUNT
11	V_ACQ_COUNT
12	V_ACQ_COUNT
13	V_ACQ_COUNT
14	V_ACQ_COUNT
15	V_ACQ_COUNT
16	V_ACQ_COUNT
17	V_ACQ_COUNT
18	V_ACQ_COUNT
19	V_ACQ_COUNT
20	V_ACQ_COUNT
21	V_ACQ_COUNT
22	V_ACQ_COUNT
23	V_ACQ_COUNT
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	V_ACQ_COUNT_CLR_MODE
29	V_ACQ_COUNT_CLR_MODE
30	V_ACQ_COUNT_UPD_MODE
31	V_ACQ_COUNT_UPD_MODE

V_ACQ_COUNT R/W, V_ACQUIRED[23..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Returns the total number of volumes (frame sequence) acquired since the last reset of this register. The behavior of this register when it reaches it maximum value depends on the register V_ACQ_COUNT_CLEAR_MODE. This register can be written to 0 by software at any time.

V_ACQ_COUNT_CLR_MODE R/W, V_ACQUIRED[29..28], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Controls how the V_ACQ_COUNT register is cleared.

V_ACQ_COUNT_CLR_MODE	Meaning
0 (00b)	Clear count on the start of acquisition
1 (01b)	Clear count on the start of V Window
2 (10b)	Clear count on the start of Z Window
3 (11b)	Clear count on the start of Y Window

V_ACQ_COUNT_UPD_MODE R/W, V_ACQUIRED[31..30], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Controls when the V_ACQ_COUNT register is update.

V_ACQ_COUNT_UPD_MODE	Meaning
0 (00b)	Update continuously
1 (01b)	Update end of V Window
2 (10b)	Update end of Z Window
3 (11b)	Update end of Y Window

2.21 Z_ACQUIRED

Bit	Name
0	Z_ACQ_COUNT
1	Z_ACQ_COUNT
2	Z_ACQ_COUNT
3	Z_ACQ_COUNT
4	Z_ACQ_COUNT
5	Z_ACQ_COUNT
6	Z_ACQ_COUNT
7	Z_ACQ_COUNT
8	Z_ACQ_COUNT
9	Z_ACQ_COUNT
10	Z_ACQ_COUNT
11	Z_ACQ_COUNT
12	Z_ACQ_COUNT
13	Z_ACQ_COUNT
14	Z_ACQ_COUNT
15	Z_ACQ_COUNT
16	Z_ACQ_COUNT
17	Z_ACQ_COUNT
18	Z_ACQ_COUNT
19	Z_ACQ_COUNT
20	Z_ACQ_COUNT
21	Z_ACQ_COUNT
22	Z_ACQ_COUNT
23	Z_ACQ_COUNT
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Z_ACQ_COUNT_CLR_MODE
29	Z_ACQ_COUNT_CLR_MODE
30	Z_ACQ_COUNT_UPD_MODE
31	Z_ACQ_COUNT_UPD_MODE

Z_ACQ_COUNT R/W, Z_ACQUIRED[23..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Returns the total number of frames acquired since the last reset of this register. The behavior of this register when it reaches it maximum value depends on the register Z_ACQ_COUNT_CLEAR_MODE. This register can be written to 0 by software at any time.

Z_ACQ_COUNT_CLR_MODE R/W, Z_ACQUIRED[29..28], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Controls how the Z_ACQ_COUNT register is cleared.

Z_ACQ_COUNT_CLR_MODE	Meaning
0 (00b)	Clear count on the start of acquisition
1 (01b)	Clear count on the start of V Window
2 (10b)	Clear count on the start of Z Window
3 (11b)	Clear count on the start of Y Window

Z_ACQ_COUNT_UPD_MODE R/W, Z_ACQUIRED[31..30], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Controls when the Z_ACQ_COUNT register is update.

Z_ACQ_COUNT_UPD_MODE	Meaning
0 (00b)	Update continuously
1 (01b)	Update end of V Window
2 (10b)	Update end of Z Window
3 (11b)	Update end of Y Window

2.22 Y_ACQUIRED

Bit	Name
0	Y_ACQ_COUNT
1	Y_ACQ_COUNT
2	Y_ACQ_COUNT
3	Y_ACQ_COUNT
4	Y_ACQ_COUNT
5	Y_ACQ_COUNT
6	Y_ACQ_COUNT
7	Y_ACQ_COUNT
8	Y_ACQ_COUNT
9	Y_ACQ_COUNT
10	Y_ACQ_COUNT
11	Y_ACQ_COUNT
12	Y_ACQ_COUNT
13	Y_ACQ_COUNT
14	Y_ACQ_COUNT
15	Y_ACQ_COUNT
16	Y_ACQ_COUNT
17	Y_ACQ_COUNT
18	Y_ACQ_COUNT
19	Y_ACQ_COUNT
20	Y_ACQ_COUNT
21	Y_ACQ_COUNT
22	Y_ACQ_COUNT
23	Y_ACQ_COUNT
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Y_ACQ_COUNT_CLR_MODE
29	Y_ACQ_COUNT_CLR_MODE
30	Y_ACQ_COUNT_UPD_MODE
31	Y_ACQ_COUNT_UPD_MODE

Y_ACQ_COUNT R/W, Y_ACQUIRED[23..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Returns the total number of lines acquired since the last reset of this register. The behavior of this register when it reaches it maximum value depends on the register Y_ACQ_COUNT_CLEAR_MODE. This register can be written to 0 by software at any time.

Y_ACQ_COUNT_CLR_MODE R/W, Y_ACQUIRED[29..28], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Controls how the Y_ACQ_COUNT register is cleared.

Y_ACQ_COUNT_CLR_MODE	Meaning
0 (00b)	Clear count on the start of acquisition
1 (01b)	Clear count on the start of V Window
2 (10b)	Clear count on the start of Z Window
3 (11b)	Clear count on the start of Y Window

Y_ACQ_COUNT_UPD_MODE R/W, Y_ACQUIRED[31..30], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Controls when the Y_ACQ_COUNT register is update.

Y_ACQ_COUNT_UPD_MODE	Meaning
0 (00b)	Update continuously
1 (01b)	Update end of V Window
2 (10b)	Update end of Z Window
3 (11b)	Update end of Y Window

2.23 X_ACQUIRED

Bit	Name
0	X_ACQ_COUNT
1	X_ACQ_COUNT
2	X_ACQ_COUNT
3	X_ACQ_COUNT
4	X_ACQ_COUNT
5	X_ACQ_COUNT
6	X_ACQ_COUNT
7	X_ACQ_COUNT
8	X_ACQ_COUNT
9	X_ACQ_COUNT
10	X_ACQ_COUNT
11	X_ACQ_COUNT
12	X_ACQ_COUNT
13	X_ACQ_COUNT
14	X_ACQ_COUNT
15	X_ACQ_COUNT
16	X_ACQ_COUNT
17	X_ACQ_COUNT
18	X_ACQ_COUNT
19	X_ACQ_COUNT
20	X_ACQ_COUNT
21	X_ACQ_COUNT
22	X_ACQ_COUNT
23	X_ACQ_COUNT
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	X_ACQ_COUNT_CLR_MODE
29	X_ACQ_COUNT_CLR_MODE
30	X_ACQ_COUNT_UPD_MODE
31	X_ACQ_COUNT_UPD_MODE

X_ACQ_COUNT R/W, X_ACQUIRED[23..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Returns the total number of 16-byte words acquired since the last reset of this register. The behavior of this register when it reaches it maximum value depends on the register X_ACQ_COUNT_CLEAR_MODE. This register can be written to 0 by software at any time.

X_ACQ_COUNT_CLR_MODE R/W, X_ACQUIRED[29..28], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Controls how the X_ACQ_COUNT register is cleared.

X_ACQ_COUNT_CLEAR_MODE	Meaning
0 (00b)	Clear count on the start of acquisition
1 (01b)	Clear count on the start of Z Window
2 (10b)	Clear count on the start of Y Window
3 (11b)	Clear count on the start of X Window

X_ACQ_COUNT_UPD_MODE R/W, X_ACQUIRED[31..30], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Controls when the X_ACQ_COUNT register is update.

X_ACQ_COUNT_UPD_MODE	Meaning
0 (00b)	Update continuously
1 (01b)	Update end of Z Window
2 (10b)	Update end of Y Window
3 (11b)	Update end of X Window

2.24 CON489

Bit	Name
0	INT_Y_ACQUIRED
1	INT_X_ACQUIRED
2	INT_Z_ACQUIRED
3	Reserved
4	INT_ENC_B
5	INT_ENC_A
6	INT_TRIG
7	INT_Y_START
8	INT_X_START
9	INT_Z_START
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	INT_BM_ERROR
27	INT_AE_LOSS_OF_SYNC
28	INT_PCIE_PKT_DROPPED
29	INT_Y_ACQUIRED_LEGACY
30	Reserved
31	Reserved

INT_Y_ ACQUIRED R/W, CON489[0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Y window closed interrupt.

INT_X_ ACQUIRED R/W, CON489[1], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

X window closed interrupt.

INT_Z_ ACQUIRED R/W, CON489[2], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Z window closed interrupt..

INT_ENC_B R/W, CON489[4], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Encoder B interrupt.

INT_ENC_A R/W, CON489[5], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Encoder A interrupt.

INT_TRIG R/W, CON489[6], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Trigger interrupt.

INT_Y_START R/W, CON489[7], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Start of Y Window interrupt.

INT_X_START R/W, CON489[8], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Start of X Window interrupt.

INT_Z_START R/W, CON489[9], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Start of Z Window interrupt.

INT_BM_ERROR R/W, CON489[26], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Buffer manager interrupt.

INT_AE_LOSS_OF_SYNC R/W, CON489[27], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Loss of sync in the Acquisition Engine interrupt.

INT_PCIE_PKT_DROPPED R/W, CON489[28], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

PCIe packet dropped interrupt.

INT_Y_ACQUIRED_LEGACY R/W, CON489[29], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Copy of INT_Y_ACQUIRED.

2.25 CON490

Bit	Name
0	Reserved
1	Reserved
2	Reserved
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	INT_ANY
8	ENINT_ALL
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

INT_ANY RO, CON490[7], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

T is at least on active interrupt on the board.

ENINT_ALL R/W, CON490[8], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Set to 1 to enable board interrupts.

2.26 CON548

Bit	Name
0	INT_Y_ACQUIRED_M
1	INT_X_ACQUIRED_M
2	INT_Z_ACQUIRED_M
3	Reserved
4	INT_ENC_B_M
5	INT_ENC_A_M
6	INT_TRIG_M
7	INT_Y_START_M
8	INT_X_START_M
9	INT_Z_START_M
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	INT_BM_ERROR_M
27	INT_AE_LOSS_OF_SYNC_M
28	INT_PCIE_PKT_DROPPED_M
29	INT_Y_ACQUIRED_LEGACY_M
30	Reserved
31	Reserved

INT_Y_ACQUIRED_M R/W, CON548[0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

INT_Y_ACQUIRED mask.

INT_X_ACQUIRED_M R/W, CON548[1], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

INT_X_ACQUIRED mask.

INT_Z_ACQUIRED_M R/W, CON548[2], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

INT_Z_ACQUIRED mask.

INT_ENC_B_M R/W, CON548[4], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

INT_ENC_B mask.

INT_ENC_A_M R/W, CON548[5], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

INT_ENC_A mask.

INT_TRIG_M R/W, CON548[6], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

INT_TRIG mask.

INT_Y_START_M R/W, CON548[7], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

INT_Y_START mask.

INT_X_START_M R/W, CON548[8], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

INT_X_START mask.

INT_Z_START_M R/W, CON548[9], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

INT_Z_START mask.

INT_BM_ERROR_M R/W, CON548[26], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

INT_BM_ERROR mask.

INT_AE_LOSS_OF_SYNC_M R/W, CON548[27], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

INT_AE_LOSS_OF_SYNC mask.

INT_PCIE_PKT_DROPPED_M R/W, CON548[28], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

INT_PCIE_PKT_DROPPED mask.

INT_Y_ACQUIRED_LEGACY_M R/W, CON548[29], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

INT_Y_ACQUIRED_LEGACY mask.

2.27 CON549

Bit	Name
0	INT_Y_ACQUIRED_WP
1	INT_X_ACQUIRED_WP
2	INT_Z_ACQUIRED_WP
3	Reserved
4	INT_ENC_B_WP
5	INT_ENC_A_WP
6	INT_TRIG_WP
7	INT_Y_START_WP
8	INT_X_START_WP
9	INT_Z_START_WP
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	INT_BM_ERROR_WP
27	INT_AE_LOSS_OF_SYNC_WP
28	INT_PCIE_PKT_DROPPED_WP
29	INT_Y_ACQUIRED_LEGACY_WP
30	Reserved
31	Reserved

INT_Y_ACQUIRED_WP R/W, CON549[0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

INT_Y_ACQUIRED write protect.

INT_X_ACQUIRED_WP R/W, CON549[1], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

INT_X_ACQUIRED write protect.

INT_Z_ACQUIRED_WP R/W, CON549[2], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

INT_Z_ACQUIRED write protect.

INT_ENC_B_WP R/W, CON549[4], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

INT_ENC_B write protect.

INT_ENC_A_WP R/W, CON549[5], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

INT_ENC_A write protect.

INT_Y_START_WP R/W, CON548[7], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

INT_Y_START write protect.

INT_X_START_WP R/W, CON548[8], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

INT_X_START write protect.

INT_Z_START_WP R/W, CON548[9], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

INT_Z_START write protect.

INT_TRIG_WP R/W, CON549[6], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

INT_TRIG write protect.

INT_BM_ERROR_WP R/W, CON549[26], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

INT_BM_ERROR write protect.

INT_AE_LOSS_OF_SYNC_WP R/W, CON549[27], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

INT_AE_LOSS_OF_SYNC write protect.

INT_PCIE_PKT_DROPPED_WP R/W, CON549[28], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

INT_PCIE_PKT_DROPPED write protect.

INT_Y_ACQUIRED_LEGACY_WP R/W, CON549[29], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

INT_Y_ACQUIRED_LEGACY write protect.

2.28 SF_DIM

Bit	Name
0	SF_HEIGHT
1	SF_HEIGHT
2	SF_HEIGHT
3	SF_HEIGHT
4	SF_HEIGHT
5	SF_HEIGHT
6	SF_HEIGHT
7	SF_HEIGHT
8	SF_HEIGHT
9	SF_HEIGHT
10	SF_HEIGHT
11	SF_HEIGHT
12	SF_HEIGHT
13	SF_HEIGHT
14	SF_HEIGHT
15	SF_HEIGHT
16	SF_WIDTH
17	SF_WIDTH
18	SF_WIDTH
19	SF_WIDTH
20	SF_WIDTH
21	SF_WIDTH
22	SF_WIDTH
23	SF_WIDTH
24	SF_WIDTH
25	SF_WIDTH
26	SF_WIDTH
27	SF_WIDTH
28	SF_WIDTH
29	SF_WIDTH
30	SF_WIDTH
31	SF_WIDTH

SF_HEIGHT R/W, SF_DIM[15..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The height (in lines) of the Synthetic Frame (internally generated synthetic image).

SF_WIDTH R/W, SF_DIM[31..16], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The width of the Synthetic frame. Units are 16 byte chunks.

2.29 SF_CON

Bit	Name
0	SF_RUN_LEVEL
1	SF_RUN_LEVEL
2	SF_STATE
3	SF_STATE
4	SF_MODE
5	SF_MODE
6	Reserved
7	SF_LINE_SCAN
8	SF_INIT_BYTE
9	SF_INIT_BYTE
10	SF_INIT_BYTE
11	SF_INIT_BYTE
12	SF_INIT_BYTE
13	SF_INIT_BYTE
14	SF_INIT_BYTE
15	SF_INIT_BYTE
16	SF_X_GAP
17	SF_X_GAP
18	SF_X_GAP
19	SF_X_GAP
20	SF_Y_GAP
21	SF_Y_GAP
22	SF_Y_GAP
23	SF_Y_GAP
24	SF_Z_GAP
25	SF_Z_GAP
26	SF_Z_GAP
27	SF_Z_GAP
28	SF_INC_X
29	SF_INC_Y
30	SF_INC_Z
31	Reserved

SF_RUN_LEVEL R/W, SF_CON[1..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The register controls the Synthetic Frame generator.

SF_RUN_LEVEL	Meaning/Command
0	Idle
1	Run
2	Abort
3	Reserved

SF_STATE RO, SF_CON[3..2], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register can be used to check the current state of the Synthetic Frame generator.

SF_MODE R/W, SF_CON[5..4], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register controls if the Synthetic Frame generator is in free-running or triggered mode.

SF_MODE	Meaning
0	Free run
1	Triggered
2	Reserved
3	Reserved

SF_LINE_SCAN R/W, SF_CON[7], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Setting SF_LINE_SCAN to one will put the Synthetic Frame generator in line scan mode .

SF_INIT_BYTE R/W, SF_CON[15..8], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The value of the first 8-bit pixel in the synthetic frame.

SF_X_GAP R/W, SF_CON[19..16], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The number of pixels between lines. Units are 16 byte chunks.

SF_Y_GAP R/W, SF_CON[23..20], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The number of lines between frames.

SF_Z_GAP R/W, SF_CON[27..24], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The number of frames between volumes.

SF_INC_X R/W, SF_CON[28], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The amount to increment the grey scale output value every pixel.

SF_INC_Y R/W, SF_CON[29], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The amount to increment the grey scale output value every line.

SF_INC_Z R/W, SF_CON[30], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The amount to increment the grey scale output value every frame.

2.30 IMAGE_STAMP_CTRL

Bit	Name
0	IS_ALL_LINES
1	Reserved
2	Reserved
3	Reserved
4	IS_MODE
5	IS_MODE
6	IS_MODE
7	IS_MODE
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

IS_ALL_LINES R/W, IMAGE_STAMP_CTRL[0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Set this bit to 1 to have image stamping on every line in the frame. If it is 0, then just the first line in the frame will be stamped.

IS_ALL_LINES	Meaning/Command
0	Timestamp just first line of frame
1	Timestamp every line

IS_MODE R/W, IMAGE_STAMP_CTRL[7..4], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register controls the boards image stamping feature. Incoming images can be stamped with metadata. The stamping overwrites the first 4 to12 bytes of image (pixel) data depending on value of this register. Use the bit IS_ALL_LINES to control if the stamping occurs on just the first line of the frame, or every line in the frame..

IS_MODE	Metadata
0 (0000b)	Image data is not stamped
1 (0001b)	32-bits, [31..0] current encoder count
2 (0010b)	64 bits, [63..56] frame counter, [55..0] on-board time-stamp
3 (0011b)	96 bits, [95..88] frame counter, [87..32] on-board timestamp, [31..0] current encoder count
4 (0100b)	96 bits, [95..0] current encoder count, each byte repeated 3 times, so the value can be read easily in RGB images
5 (0101b)	Reserved
6 (0110b)	Reserved
7 (0111b)	Reserved
8 (1000b)	Reserved
9 (1001b)	Reserved
10 (1010b)	Reserved
11 (1011b)	Reserved
12 (1100b)	Reserved
13 (1101b)	Reserved
14 (1110b)	Reserved
15 (111b)	Reserved

The StreamSync Buffer Manager

Chapter 3

3.1 Introduction

The StreamSync system consists of an Acquisition Engine and a Buffer Manager. The StreamSync system was first released on the Cyton-CXP and is a departure from previous BitFlow frame grabbers. The StreamSync system is a start-from-scratch complete redesign of the acquisition and DMA parts of a frame grabber. BitFlow used its years of experience in this area to design a next generation, super efficient capture system.

The StreamSync system is supported on the Aon-CXP, the Axion-CL, the Claxon-CXP, Claxon-FXP and the Cyton-CXP.

From a software perspective, the StreamSync system is compatible with the previous BitFlow products. However, digging deeper, these new system have a lot more power and flexibility. These new features will be described in the following sections.

The StreamSync system has many improvements over previous systems. The main improvements are:

- Efficient support for variable sized images with fast context switches between frames
- Per frame control of acquisition properties (AOI specifically)
- Hardware control of image sequencing
- Enhanced debug capabilities
- Efficient support for on-demand buffer allocation (GenIcam model)
- Gracefully recovery from dropped packets (either on the input side or the DMA side)

This chapter describes the StreamSync Buffer Manager while the previous chapter describes the StreamSync Acquisition Engine.

3.2 The Buffer Manager Details

The Buffer Manager interacts with a remote, software managed, set of Scatter Gather DMA lists. A single Scatter Gather DMA list is called a QTab. A QTab is made of individual DMA instructions (descriptors) called Quads. One Quad contains the information to DMA one contiguous chunk of data from the board to host memory. The Buffer Manager reads in and caches QTabs and the associated Quads. It makes the cached Quads and QTabs available to the Acquisition Engine in queued order. The Buffer Manager works independently of the Acquisition Engine and can be throttled by software.

The Buffer Manager and Acquisition Engine are designed to work asynchronously from each other. The Buffer Manager is capable of reading in Quads from the remote QTab while the Acquisition Engine is Running/Stopping/Aborting/or Stopped. If the local Buffer Cache fills and the Acquisition Engine is not currently consuming Quads, the Buffer Manager simply waits until room becomes available and pauses loading Quads from the remote QTab. Likewise, the Acquisition Engine is capable of acquiring frames as long as it is running and has Quad available to work on. If no Quads are available it will simply wait for more to become available from the Buffer Manager. The Acquisition Engine can accept commands of Stop/Abort/Start, all while the Buffer Manager is running independently.

The starting, stopping, and restarting of the Acquisition Engine and Buffer Manager, however, does require some synchronization. The Buffer Manager pre-fetches Quad and Quads for efficiency. This built up pipeline and caching structure requires the Acquisition Engine to be in the Stopped state before the Buffer Manager can be safely flushed. Flushing of the Buffer Manager happens when the user wants to completely shut down the StreamSync Acquisition Engine or simply start acquiring to a new QTab.

3.3 CON485 Register

Bit	Name
0	FIRST_QUAD_PTR_LO
1	FIRST_QUAD_PTR_LO
2	FIRST_QUAD_PTR_LO
3	FIRST_QUAD_PTR_LO
4	FIRST_QUAD_PTR_LO
5	FIRST_QUAD_PTR_LO
6	FIRST_QUAD_PTR_LO
7	FIRST_QUAD_PTR_LO
8	FIRST_QUAD_PTR_LO
9	FIRST_QUAD_PTR_LO
10	FIRST_QUAD_PTR_LO
11	FIRST_QUAD_PTR_LO
12	FIRST_QUAD_PTR_LO
13	FIRST_QUAD_PTR_LO
14	FIRST_QUAD_PTR_LO
15	FIRST_QUAD_PTR_LO
16	FIRST_QUAD_PTR_LO
17	FIRST_QUAD_PTR_LO
18	FIRST_QUAD_PTR_LO
19	FIRST_QUAD_PTR_LO
20	FIRST_QUAD_PTR_LO
21	FIRST_QUAD_PTR_LO
22	FIRST_QUAD_PTR_LO
23	FIRST_QUAD_PTR_LO
24	FIRST_QUAD_PTR_LO
25	FIRST_QUAD_PTR_LO
26	FIRST_QUAD_PTR_LO
27	FIRST_QUAD_PTR_LO
28	FIRST_QUAD_PTR_LO
29	FIRST_QUAD_PTR_LO
30	FIRST_QUAD_PTR_LO
31	FIRST_QUAD_PTR_LO

FIRST_QUAD_PTR_LO R/W, CON28[31..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This is the low word of the 64-bit address of the first DMA scatter-gather instruction in a chain of instructions.

3.4 CON486 Register

Bit	Name
0	FIRST_QUAD_PTR_HI
1	FIRST_QUAD_PTR_HI
2	FIRST_QUAD_PTR_HI
3	FIRST_QUAD_PTR_HI
4	FIRST_QUAD_PTR_HI
5	FIRST_QUAD_PTR_HI
6	FIRST_QUAD_PTR_HI
7	FIRST_QUAD_PTR_HI
8	FIRST_QUAD_PTR_HI
9	FIRST_QUAD_PTR_HI
10	FIRST_QUAD_PTR_HI
11	FIRST_QUAD_PTR_HI
12	FIRST_QUAD_PTR_HI
13	FIRST_QUAD_PTR_HI
14	FIRST_QUAD_PTR_HI
15	FIRST_QUAD_PTR_HI
16	FIRST_QUAD_PTR_HI
17	FIRST_QUAD_PTR_HI
18	FIRST_QUAD_PTR_HI
19	FIRST_QUAD_PTR_HI
20	FIRST_QUAD_PTR_HI
21	FIRST_QUAD_PTR_HI
22	FIRST_QUAD_PTR_HI
23	FIRST_QUAD_PTR_HI
24	FIRST_QUAD_PTR_HI
25	FIRST_QUAD_PTR_HI
26	FIRST_QUAD_PTR_HI
27	FIRST_QUAD_PTR_HI
28	FIRST_QUAD_PTR_HI
29	FIRST_QUAD_PTR_HI
30	FIRST_QUAD_PTR_HI
31	FIRST_QUAD_PTR_HI

FIRST_QUAD_PTR_HI R/W, CON29[31..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This is the high word of the 64-bit address of the first DMA scatter-gather instruction in a chain of instructions.

3.5 BUF_MGR_CON

Bit	Name
0	BM_RUN_LEVEL
1	BM_RUN_LEVEL
2	BM_RUN_LEVEL
3	BM_RUN_LEVEL
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	CURR_FETCH_SIZE
25	CURR_FETCH_SIZE
26	CURR_FETCH_SIZE
27	CURR_FETCH_SIZE
28	MAX_FETCH_SIZE
29	MAX_FETCH_SIZE
30	MAX_FETCH_SIZE
31	MAX_FETCH_SIZE

BM_RUN_LEVEL R/W, BUF_MGR_CON[3..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This is the main control for starting/stopping the Buffer Manager.

BM_RUN_LEVEL	Meaning
0 (0000b)	Idle - The Buffer Manager is not moving data
1 (0001b)	Run - The Buffer Manger will start to move data
2 (0010b)	Abort - Abort DMA and go to Idle
3 (0011b)	

CURR_FETCH_SIZE RO, BUF_MGR_CON[27..24], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This is the number of Quads that will be fetched at a time by the Buffer Manager. A large read is issued over the PCIe bus to read all these Quads at one time.

CURR_FETCH_SIZE	Meaning
0 (0000b)	1 Quad
1 (0001b)	2 Quads
2 (0010b)	4 Quads
3 (0011b)	8 Quads
15 (1111b)	32K Quads

MAX_FETCH_SIZE RO, BUF_MGR_CON[31..28], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This is the maximum number of Quads that can be fetched as a group by the Buffer Manager. The value in this register is derived as a function of the maximum PCIe read request size set by PCI enumeration.

3.6 BUF_MGR_TIMEOUT

Bit	Name
0	QUAD_COMPLETE_TIMEOUT
1	QUAD_COMPLETE_TIMEOUT
2	QUAD_COMPLETE_TIMEOUT
3	QUAD_COMPLETE_TIMEOUT
4	QUAD_COMPLETE_TIMEOUT
5	QUAD_COMPLETE_TIMEOUT
6	QUAD_COMPLETE_TIMEOUT
7	QUAD_COMPLETE_TIMEOUT
8	QUAD_COMPLETE_TIMEOUT
9	QUAD_COMPLETE_TIMEOUT
10	QUAD_COMPLETE_TIMEOUT
11	QUAD_COMPLETE_TIMEOUT
12	QUAD_COMPLETE_TIMEOUT
13	QUAD_COMPLETE_TIMEOUT
14	QUAD_COMPLETE_TIMEOUT
15	QUAD_COMPLETE_TIMEOUT
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	DISABLE_TIMEOUT

QUAD_COMPLETE_TIMEOUT

R/W, BUF_MGR_TIMEOUT[15..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The maximum amount of time to wait for a Quad completion. Units are 4 nanoseconds. Writable only when BM_STATE is Idle.

DISABLE_TIMEOUT

R/W, BUF_MGR_TIMEOUT[31], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Setting this bit to 1 will disable the Quad completion timeout mechanism. The Buffer Manager will wait an infinite amount of time for a Quad completion to return. For debug only. Writable only when BM_STATE is Idle.

3.7 BOARD_CONFIG

Bit	Name
0	SW
1	SW
2	Reserved
3	Reserved
4	CPLD_MODE
5	CPLD_MODE
6	CPLD_MODE
7	CPLD_MODE
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	CPLD_STRAP
13	CPLD_STRAP
14	CPLD_STRAP
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

SW RO, BOARD_CONFIG[1..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The current value of the on board switch SW1.

CPLD_MODE RO, BOARD_CONFIG[7..4], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The current value of switch S3. This switch controls the firmware bank that the FPGA boots from.

CPLD_STRAP RO, BOARD_CONFIG[14..12], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The current value of the three on board straps.

3.8 PACKETS_SENT_STATUS

Bit	Name
0	NUM_PACKETS_SENT
1	NUM_PACKETS_SENT
2	NUM_PACKETS_SENT
3	NUM_PACKETS_SENT
4	NUM_PACKETS_SENT
5	NUM_PACKETS_SENT
6	NUM_PACKETS_SENT
7	NUM_PACKETS_SENT
8	NUM_PACKETS_SENT
9	NUM_PACKETS_SENT
10	NUM_PACKETS_SENT
11	NUM_PACKETS_SENT
12	NUM_PACKETS_SENT
13	NUM_PACKETS_SENT
14	NUM_PACKETS_SENT
15	NUM_PACKETS_SENT
16	NUM_PACKETS_DROP
17	NUM_PACKETS_DROP
18	NUM_PACKETS_DROP
19	NUM_PACKETS_DROP
20	NUM_PACKETS_DROP
21	NUM_PACKETS_DROP
22	NUM_PACKETS_DROP
23	NUM_PACKETS_DROP
24	NUM_PACKETS_DROP
25	NUM_PACKETS_DROP
26	NUM_PACKETS_DROP
27	NUM_PACKETS_DROP
28	NUM_PACKETS_DROP
29	NUM_PACKETS_DROP
30	NUM_PACKETS_DROP
31	NUM_PACKETS_DROP

NUM_PACKETS_SENT RO, PACKETS_SENT_STATUS[15..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The register indicates the number of PCIe packets that the Buffer Manager has sent across the PCIe bus. This register rolls over to 0 at 0xffff.

NUM_PACKETS_DROP RO, PACKETS_SENT_STATUS[31..16], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register indicates the number of PCIe packets that the buffer Manager was not able to send across the PCIe bus because the PCIe bus was busy. These packets are dropped, but the corresponding Quads are also "consumed". This means that the Buffer Manager still stays synchronized and any subsequent packets will be DMA to their correct locations.

3.9 QUADS_USED_STATUS

Bit	Name
0	NUM_QUADS_USED
1	NUM_QUADS_USED
2	NUM_QUADS_USED
3	NUM_QUADS_USED
4	NUM_QUADS_USED
5	NUM_QUADS_USED
6	NUM_QUADS_USED
7	NUM_QUADS_USED
8	NUM_QUADS_USED
9	NUM_QUADS_USED
10	NUM_QUADS_USED
11	NUM_QUADS_USED
12	NUM_QUADS_USED
13	NUM_QUADS_USED
14	NUM_QUADS_USED
15	NUM_QUADS_USED
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

NUM_QUADS_USED RO, QUADS_USED_STATUS[15..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register indicates the number of Quads that have been “consumed” by the Buffer Manager. This register rolls over to 0 at 0xffff.

3.10 QTABS_USED_STATUS

Bit	Name
0	NUM_QTABS_USED
1	NUM_QTABS_USED
2	NUM_QTABS_USED
3	NUM_QTABS_USED
4	NUM_QTABS_USED
5	NUM_QTABS_USED
6	NUM_QTABS_USED
7	NUM_QTABS_USED
8	NUM_QTABS_USED
9	NUM_QTABS_USED
10	NUM_QTABS_USED
11	NUM_QTABS_USED
12	NUM_QTABS_USED
13	NUM_QTABS_USED
14	NUM_QTABS_USED
15	NUM_QTABS_USED
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

NUM_QTABS_USED RO, QTABS_USED_STATUS[15..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register indicates the number of QTabs that have been “consumed” by the Buffer Manager. This register rolls over to 0 at 0xffff.

3.11 PKT_STAT

Bit	Name
0	PKT_STATE
1	PKT_STATE
2	Reserved
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	NO_QUAD_AVAIL
9	VIDEO_DROPPED
10	QUAD_DROPPED
11	Reserved
12	NEW_FRAME_RESYNC
13	RD_ON_EMPTY
14	WR_ON_FULL
15	Reserved
16	PKT_FLUSH_ENABLE
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

PKT_STATE RO, PKT_STAT[1..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Current state of the DMA engine.

PKT_STATE	Meaning
0 (00b)	PKT_SYNC - Synchronizing DMA descriptors with video
1 (01b)	PKT_HDR - Generating PCIe header
2 (10b)~	PKT_DAT - Placing data in PCIe packet
3 (11b)	Reserved

NO_QUAD_AVAIL RO, PKT_STAT[8], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

StreamSync DMA has data to transmit but no descriptor (effectively no valid place to send data). This indicates a problem with fetching descriptors.

VIDEO_DROPPED RO, PKT_STAT[9], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Can occur during PKT_SYNC as video from acquisition engine is dropped in order to resynchronize.

QUAD_DROPPED RO, PKT_STAT[10], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Similar to VIDEO_DROPPED but indicates quad was dropped during re-sync process when PKT_STATE equals PKT_SYNC.

NEW_FRAME_RESYNC RO, PKT_STAT[12], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Reserved.

RD_ON_EMPTY RO, PKT_STAT[13], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

FIFO underflow in Packet Engine.

WR_ON_FULL R/W, PKT_STAT[14], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

FIFO overflow in Packet Engine.

PKT_FLUSH_ENABLE R/W, PKT_STAT[16], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

DMA tries to send as large as packets as possible for efficiency. Data is collected in a FIFO until certain size rules are met. However, sometimes no more data will be coming (end of frame). In this case, a timeout forces the Packet Engine to transmit the remaining data. PKT_FLUSH_ENABLE = 1 indicates that this has taken place.

3.12 QUADS_LOADED_STATUS

Bit	Name
0	NUM_QUADS_LOADED
1	NUM_QUADS_LOADED
2	NUM_QUADS_LOADED
3	NUM_QUADS_LOADED
4	NUM_QUADS_LOADED
5	NUM_QUADS_LOADED
6	NUM_QUADS_LOADED
7	NUM_QUADS_LOADED
8	NUM_QUADS_LOADED
9	NUM_QUADS_LOADED
10	NUM_QUADS_LOADED
11	NUM_QUADS_LOADED
12	NUM_QUADS_LOADED
13	NUM_QUADS_LOADED
14	NUM_QUADS_LOADED
15	NUM_QUADS_LOADED
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

NUM_QUADS_LOADED RO, QUADS_LOADED_STATUS[15..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register indicates the number of Quads that have been loaded by the Buffer Manager. This register will roll over to 0 at 0xffff.

3.13 QTABS_LOADED_STATUS

Bit	Name
0	NUM_QTABS_LOADED
1	NUM_QTABS_LOADED
2	NUM_QTABS_LOADED
3	NUM_QTABS_LOADED
4	NUM_QTABS_LOADED
5	NUM_QTABS_LOADED
6	NUM_QTABS_LOADED
7	NUM_QTABS_LOADED
8	NUM_QTABS_LOADED
9	NUM_QTABS_LOADED
10	NUM_QTABS_LOADED
11	NUM_QTABS_LOADED
12	NUM_QTABS_LOADED
13	NUM_QTABS_LOADED
14	NUM_QTABS_LOADED
15	NUM_QTABS_LOADED
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

NUM_QTABS_LOADED RO, QTABS_LOADED_STATUS[15..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register indicates the number of QTabs that have been loaded by the Buffer Manager. This register will roll over to 0 at 0xffff.

3.14 BUF_MGR_STATUS

Bit	Name
0	BM_STATE
1	BM_STATE
2	BM_STATE
3	Reserved
4	CPL_STATUS
5	CPL_STATUS
6	CPL_STATUS
7	Reserved
8	BM_QUADS_CACHED
9	BM_QUADS_CACHED
10	BM_QUADS_CACHED
11	BM_QUADS_CACHED
12	BM_QUADS_CACHED
13	BM_QUADS_CACHED
14	BM_QUADS_CACHED
15	BM_QUADS_CACHED
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	DST_ADDR_ERROR_LSB
21	NEXT_ADDR_ERROR_LSB
22	SIZE_ERROR_LSB
23	SIZE_ERROR_MSB
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	CPL_ERROR
29	QUAD_NUM_MISMATCH
30	QUAD_FIFO_OVERFLOW
31	QUAD_TIMEOUT_DETECTED

BM_STATE RO, BUF_MGR_STATUS[2..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Returns the current state of the Buffer Manager.

BM_STATE	Meaning
0 (0000b)	Idle - The buffer manager is not current active
1 (0001b)	Active - The buffer manager is currently DMAing
2 (0010b)	Req64
3 (0011b)	Req32
4 (0100b)	Wait CPL
4 (0101b)	Parse CPL 0
6 (0110b)	Parse CPL 0
7 (0111b)	Flush

CPL_STATUS RO, BUF_MGR_STATUS[6..4], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

PCIe completion status from last received Quad.

BM_QUADS_CACHED RO, BUF_MGR_STATUS[15..8], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of QUADS currently in the cache.

DST_ADDR_ERROR_LSB RO, BUF_MGR_STATUS[20], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Quad destination address is not 16 byte aligned.

NEXT_ADDR_ERROR_LSB RO, BUF_MGR_STATUS[21], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Quad points to a next quad that is not 16-byte aligned..

SIZE_ERROR_LSB RO, BUF_MGR_STATUS[22], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Quad size is not a multiple of 16 bytes.

SIZE_ERROR_MSB RO, BUF_MGR_STATUS[23], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Quad size is > 4K.

CPL_ERROR RO, BUF_MGR_STATUS[28], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Error code received as a result of fetching a Quad. Check CPL_STATUS.

QUAD_NUM_MISMATCH RO, BUF_MGR_STATUS[29], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Actual quad number does not match expected.

QUAD_FIFO_OVERFLOW RO, BUF_MGR_STATUS[30], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Quad cache overflowed.

QUAD_TIMEOUT_DETECTED RO, BUF_MGR_STATUS[31], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Timeout waiting for a Quad completion. A different timeout value can be set in the QUAD_COMPLETE_TIMEOUT register.

3.15 PKT_CON

Bit	Name
0	MAX_PAYLOAD_USER
1	MAX_PAYLOAD_USER
2	MAX_PAYLOAD_USER
3	MAX_PAYLOAD_USER
4	MAX_PAYLOAD_PCIE
5	MAX_PAYLOAD_PCIE
6	MAX_PAYLOAD_PCIE
7	MAX_PAYLOAD_PCIE
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	DISABLE_PKT_FLUSH_TIMER
31	DISABLE_PKT_GEN

MAX_ PAYLOAD_USER RO, PKT_CON[3..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This is the maximum sized PCIe packet that will be generated by the Buffer Manager. Writes to this register of values higher than MAX_PAYLOAD_PCIE will be ignored. The coding is shown in the following table.

MAX_PAYLOAD_USER	Meaning
0 (0000b)	16 bytes
1 (0001b)	32 bytes
2 (0010b)	64 bytes
3 (0011b)	128 bytes
4 (0100b)	256 bytes
4 (0101b)	512 bytes
6 (0110b)	1024 bytes
7 (0111b)	2048 bytes
8 (1000b)	4096 bytes

MAX_ PAYLOAD_PCIE RO, PKT_CON[7..4], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This is the maximum sized PCIe write packet that can be generated by the Buffer Manager for video data. This value is set by the PCIe enumeration. The coding for this field is the same as listed under MAX_PAYLOAD_USER. Only values of 128 bytes to 4096 bytes are possible in PCIe, however, MAX_PAYLOAD_USER provides a few smaller values (16 bytes to 64 bytes) for testing purposes only. MAX_PAYLOAD_PCIE is status only and does not control internal logic. MAX_PAYLOAD_USER does control internal logic.

DISABLE_PKT_ FLUSH_TIMER R/W, PKT_CON[30], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Deactivate the timer that flushes video data to PCIe when the FIFO is inactive for an extended period.

Note: This bit is for degging purposes only.

DISABLE_PKT_ GEN R/W, PKT_CON[31], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Disable the generation of outbound PCIe video packets. This is a final stage disable. The packet is actually generated by the logic as if it were going to be transmitted. This means that all address's and other counters increment as if the packet were generated. However it is simply dropped afterwards.

Note: This bit is for debugging purposes only.

Timing Sequencer

Chapter 4

4.1 Introduction

This section covers the Timing Sequencer (TS) which is available on the Aon, Axion, Claxon and Cyton. The TS is a sophisticated programmable pulse generator. The TS takes the place of the NTG on previous models of BitFlow frame grabbers.

The TS improves on the NTG in the following ways:

- Driven by a "nice" clock frequency clock so that "normal" pulse sizes and periods can easily be reproduced (for example 1 micro second pulse every 10 milliseconds)
- Higher accuracy signals, granularity down to 100 nanoseconds
- Supports complex pulse trains of different lengths
- Provides synchronized method to switch from one pulse sequence to another
- Can be reprogrammed while being used (with some restrictions)
- Supports triggering at arbitrary points in a pulse train

4.1.1 The Auxiliary Timing Sequencer

The first Virtual Frame Grabber (VFG0) has two independent Timing Sequencers: TS and ATS. They work exactly the same, and have the same control registers. The Auxiliary Timing Sequencer registers are listed here, even though they work in the exact same way as the main timing sequencer. All of the description that follows applies to both the TS and the ATS.

Note: The Auxiliary Timing Sequencer (ATS) is only available on VFG0.

4.1.2 Description

TS Table

The TS is programmed through the TS registers. The sequence of pulses that the TS will output is programmed by building up "instructions" in the TS table. The table can hold up to 256 instructions, which can create extremely complex signals. The TS Table is programmed indirectly via address/data type registers. Once the table is programmed it can be run at any time via the TS control registers.

Building Pulses

The TS was designed to support a wide range of pulse lengths. At the same time, the TS was design to be able to create pulses of very accurate duration. The solution to these two opposing problems is to build up a pulse of a desired length via multiple sub-pulses, each sub-pulse programmed with a different granularity. The following granularities are available:

100 seconds
100 milliseconds
100 microseconds
100 nanoseconds

Each sub-pulse can have a length of 1 to 1023 units, where the units are selected from the list above.

For example, let's say you want a pulse that is 1.2345678 seconds long. This is done by programing the TS table with three sub pulse as shown below:

Entry 1: 12 * 100 milliseconds = 1.2 seconds
Entry 2: 345 * 100 microseconds = 0.0345 seconds
Entry 3: 678 * 100 nanoseconds = 0.0000678 seconds

When these three entries are "run", they are output sequentially and seamlessly, creating a single pulse of the desired length:

1.2 + 0.0345 + 0.0000678 = 1.2345678

As you can see, this system provides both a wide range of durations as well as very accurate durations.

Pulses can be either high or low (0 or 1). By programming both high pulses and low pulses any pulse train can be created.

Chaining Pulses

Pulses are chained together by a linked list. Each pulse entry has a "next" field which tells the system where in the table the next pulse should come from. This facility is used to build up complex sequences as well as looping sequences.

Triggering

Each entry in the TS table can produce one pulse. Each entry has a "condition" under which it will get executed. The conditions can be immediate. In other words, as soon as the TS gets to this entry, it immediately produces the programmed pulses. Or it can be programmed to wait for a trigger. Various trigger conditions are supported.

By adding this condition, the pulse train produced can be run in "one-shot" mode, where one trigger produces one or more pulses.

Each pulse can use one of two different trigger sources. The two trigger sources are set by the TS_TRIG_SEL_0 and TS_TRIG_SEL_1. For example, you have one pulse that is triggered by the Start Of Frame, and another pulse that is triggered by the encoder. This way you can have the board output a pulse every frame to one destination (e.g. strobe) and the other signal can output a pulse every line (e.g. to the camera). The only issue with this scenario is that it needs to be reset at the end of the frame. The need is met by the immediate jump facility described in the next paragraph.

Because it's possible to configure the TS two work off two different trigger sources, a mechanism is need to be able to force TS to jump from one area to another without CPU intervention. The Immediate Jump function has been added to help with this feature. This function has it's own trigger selector, so a third trigger source can be used to initiate the jump. The jump destination using the existing TS_IDX_JUMP bitfield that is normally used for software controlled jump. The the Immediate Jump trigger asserts, the Timing Sequencer index immediately jumps to TS_IDX_JUMP address. Normal TS operation continues from this point.

In the example above, the Immediate Jump can reset the TS so its ready to output the pulse at the start of frame. This would be done by setting the Immediate Jump trigger to the End of Frame signal, and the TS_IDX_JUMP to the location of the first pulse that is waiting for the Start of Frame.

4.2 TS_CONTROL

Bit	Name
0	TS_RUN_LEVEL
1	TS_RUN_LEVEL
2	TS_RUN_LEVEL
3	Reserved
4	TS_CT0_DEFAULT_STATE
5	TS_CT1_DEFAULT_STATE
6	TS_CT2_DEFAULT_STATE
7	TS_CT3_DEFAULT_STATE
8	TS_IMMDT_JUMP_SEL
9	TS_IMMDT_JUMP_SEL
10	TS_IMMDT_JUMP_SEL
11	TS_IMMDT_JUMP_SEL
12	TS_IMMDT_JUMP_COND
13	TS_IMMDT_JUMP_EN
14	Reserved
15	Reserved
16	TS_IDX_JUMP
17	TS_IDX_JUMP
18	TS_IDX_JUMP
19	TS_IDX_JUMP
20	TS_IDX_JUMP
21	TS_IDX_JUMP
22	TS_IDX_JUMP
23	TS_IDX_JUMP
24	TS_TRIG_SEL_1
25	TS_TRIG_SEL_1
26	TS_TRIG_SEL_1
27	TS_TRIG_SEL_1
28	TS_TRIG_SEL_0
29	TS_TRIG_SEL_0
30	TS_TRIG_SEL_0
31	TS_TRIG_SEL_0

TS_RUN_LEVEL R/W, TS_CONTROL[2..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

These bits control the operation of the TS. These bits are used to start and stop the sequencer. They can also be used to program the table to jump to a new section. Jumps are always synchronous (i.e. not immediate). Jumps will only occur from an index in the sequence that has the TS_END_OF_SEQUENCE bit set.

This bit can be read at any time in order get the current status.

The following table shows the command available for this register.

TS_RUN_LEVEL	Meaning
0 (000b)	Idle - TS is not running
1 (001b)	Run - Start running immediately from index in the TS_IDX_JUMP register
2 (010b)	Jump - Jump to index set in the TS_IDX_JUMP register next time the current index has the TS_END_OF_SEQUENCE bit set to 1
3 (011b)	Stop - Stop running the next time the current index has the TS_END_OF_SEQUENCE bit set to 1
4 (100b)	Abort - Stop running immediately

TS_CT0_DEFAULT_STATE R/W, TS_CONTROL[4], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This is the output state of CT0 when the TS is Idle.

TS_CT1_DEFAULT_STATE R/W, TS_CONTROL[5], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This is the output state of CT1 when the TS is Idle.

TS_CT2_DEFAULT_STATE R/W, TS_CONTROL[6], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This is the output state of CT2 when the TS is Idle.

TS_CT3_DEFAULT_STATE R/W, TS_CONTROL[7], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This is the output state of CT3 when the TS is Idle.

TS_IMMDT_JUMP_SEL R/W, TS_CONTROL[8..11], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

These bits select the source of the Immediate Jump function.

TS_IMMDT_JUMP_SEL	Meaning
0 (0000b)	Selected trigger (VGFx_TRIG_SEL)
1 (0001b)	Selected encoder A (VFGx_ENCA_SEL)
2 (0010b)	Selected encoder B (VFGx_ENCB_SEL)
3 (0011b)	Selected quad encoder output (VFGx_ENCQ_SEL)
4 (0100b)	Gated trigger (VGFx_TRIG_SEL gated by VFGx_ENCB_SEL)
5 (0101b)	Selected encoder divider output (VFGx_ENCDIV_SEL)
6 (0110b)	AE start of frame (Y Window Opens)
7 (0111b)	AE end of frame (Y Window Closes)
8 (1000b)	AE start of line (X Window Opens)
9 (1001b)	AE end of line (X Window Closes)
10 (1010b)	FVAL rising edge*
11 (1011b)	FVAL falling edge*
12 (1100b)	LVAL rising edge*
13 (1101b)	LVAL falling edge*
14 (1110b)	Reserved
15 (1111b)	Reserved

*Note: * FVAL and LVAL triggers are only support on the Axion models*

TS_IMMDT_JUMP_COND R/W, TS_CONTROL[12], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This controls the polarity of the Immediate Jump function.

TS_IMMDT_JUMP_COND	Meaning
0	Rising edge
1	Falling edge

TS_IMMDT_JUMP_EN R/W, TS_CONTROL[12], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Set this bit to 1 to enable Immediate Jump function.

TS_IDX_JUMP R/W, TS_CONTROL[23..16], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This is the entry that the table will start from when the TS_RUN_LEVEL register is set to Run.

This is the entry that the table will jump to (synchronously) when the TS_RUN_LEVEL register is set to Jump.

TS_TRIG_SEL_1 R/W, TS_CONTROL[27..24], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

These bits select the source of the TS trigger 1.

TS_TRIG_SEL_1	Meaning
0 (0000b)	Selected trigger (VGFx_TRIG_SEL)
1 (0001b)	Selected encoder A (VFGx_ENCA_SEL)
2 (0010b)	Selected encoder B (VFGx_ENCB_SEL)
3 (0011b)	Selected quad encoder output (VFGx_ENCQ_SEL)
4 (0100b)	Gated trigger (VGFx_TRIG_SEL gated by VFGx_ENCB_SEL)
5 (0101b)	Selected encoder divider output (VFGx_ENCDIV_SEL)
6 (0110b)	AE start of frame (Y Window Opens)
7 (0111b)	AE end of frame (Y Window Closes)
8 (1000b)	AE start of line (X Window Opens)
9 (1001b)	AE end of line (X Window Closes)
10 (1010b)	FVAL rising edge*
11 (1011b)	FVAL falling edge*
12 (1100b)	LVAL rising edge*
13 (1101b)	LVAL falling edge*
14 (1110b)	Reserved
15 (1111b)	Reserved

*Note: * FVAL and LVAL triggers are only support on the Axion models*

TS_TRIG_SEL_0 R/W, TS_CONTROL[31..28], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

These bits select the source of the TS trigger 0.

TS_TRIG_SEL_0	Meaning
0 (0000b)	Selected trigger (VGFx_TRIG_SEL)
1 (0001b)	Selected encoder A (VFGx_ENCA_SEL)
2 (0010b)	Selected encoder B (VFGx_ENCB_SEL)
3 (0011b)	Selected quad encoder output (VFGx_ENCQ_SEL)
4 (0100b)	Gated trigger (VGFx_TRIG_SEL gated by VFGx_ENCB_SEL)
5 (0101b)	Selected encoder divider output (VFGx_ENCDIV_SEL)
6 (0110b)	AE start of frame (Y Window Opens)
7 (0111b)	AE end of frame (Y Window Closes)
8 (1000b)	AE start of line (X Window Opens)
9 (1001b)	AE end of line (X Window Closes)
10 (1010b)	FVAL rising edge*
11 (1011b)	FVAL falling edge*
12 (1100b)	LVAL rising edge*
13 (1101b)	LVAL falling edge*
14 (1110b)	Reserved
15 (1111b)	Reserved

*Note: * FVAL and LVAL triggers are only support on the Axion models*

4.3 TS_TABLE_CONTROL

Bit	Name
0	TS_IDX_ACCESS
1	TS_IDX_ACCESS
2	TS_IDX_ACCESS
3	TS_IDX_ACCESS
4	TS_IDX_ACCESS
5	TS_IDX_ACCESS
6	TS_IDX_ACCESS
7	TS_IDX_ACCESS
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

TS_IDX_ACCESS R/W, TS_TABLE_CONTROL[7..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Indirect access to the TS table. Set this bitfield to the index value that you wish to modify. Access is done through via TS_TABLE_ENTRY register.

4.4 TS_TABLE_ENTRY

Bit	Name
0	TS_NEXT
1	TS_NEXT
2	TS_NEXT
3	TS_NEXT
4	TS_NEXT
5	TS_NEXT
6	TS_NEXT
7	TS_NEXT
8	Reserved
9	Reserved
10	TS_RESOLUTION
11	TS_RESOLUTION
12	TS_STATE_CT0
13	TS_STATE_CT1
14	TS_STATE_CT2
15	TS_STATE_CT3
16	TS_TRIG_SEL_SEL
17	TS_COUNT
18	TS_COUNT
19	TS_COUNT
20	TS_COUNT
21	TS_COUNT
22	TS_COUNT
23	TS_COUNT
24	TS_COUNT
25	TS_COUNT
26	TS_COUNT
27	TS_CONDITION
28	TS_CONDITION
29	TS_CONDITION
30	TS_TERMINATE
31	TS_END_OF_SEQUENCE

TS_NEXT R/W, TS_TABLE_ENTRY[7..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Index of next pulse. Only relevant if TS_TERMINATE = 0.

TS_RESOLUTION R/W, TS_TABLE_ENTRY[11..10], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The time units of this pulse. The length of this pulse is set in the register TS_COUNT. The following table shows the available resolutions.

TS_RESOLUTION	Meaning
0 (000b)	100 nanoseconds
1 (001b)	100 microseconds
2 (010b)	100 milliseconds
3 (011b)	100 seconds

TS_STATE_CT0 R/W, TS_TABLE_ENTRY[12], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The level of the CT0 signal for this pulse.

TS_STATE_CT1 R/W, TS_TABLE_ENTRY[13], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The level of the CT1 signal for this pulse.

TS_STATE_CT2 R/W, TS_TABLE_ENTRY[14], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The level of the CT2 signal for this pulse.

TS_STATE_CT3 R/W, TS_TABLE_ENTRY[15], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The level of the CT3 signal for this pulse.

TS_TRIG_SEL_SEL R/W, TS_TABLE_ENTRY[16], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

If this pulse is triggered, this bit sets which trigger will be used.

TS_TRIG_SEL_SEL	Meaning
0	The trigger selected by TS_TRIG_SEL_0
1	The trigger selected by TS_TRIG_SEL_1

TS_COUNT R/W, TS_TABLE_ENTRY[26..17], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The length of this pulse. The units for the length are set in the TS_RESOLUTION register.

TS_CONDITION R/W, TS_TABLE_ENTRY[29..27], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register is used to control the conditions under which this pulse will be output. The following table shows the options for this bitfield.

TS_CONDITION	Condition when pulse is output
0 (000b)	Immediate
1 (001b)	Rising edge of trigger
2 (010b)	Falling edge of trigger
3 (011b)	Trigger high
4 (100b)	Trigger low
5 (101b)	Both rising and falling edge of trigger

TS_TERMINATE R/W, TS_TABLE_ENTRY[30], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

When this bit is set to 1, the table will stop running after the current pulse is output. The TS_RUN_LEVEL bitfield will then read back idle.

When this bit is set to 0, the table will jump to the index set in the TS_NEXT bitfield after the current pulse is finished.

TS_END_OF_SEQUENCE R/W, TS_TABLE_ENTRY[31], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

If this bit is set to 1 and TS_RUN_LEVEL is set to Jump, the TS will jump to the index set in the TS_INDX_JUMP bitfield after the current pulse is output. This bit allows for synchronous switching between one section of the table and another section.

If the TS_RUN_LEVEL bitfield is not set to Jump, then this bitfield will have no effect.

If this bit is set to 0, the TS will not jump from this index.

4.5 ATS_CONTROL

Bit	Name
0	ATS_RUN_LEVEL
1	ATS_RUN_LEVEL
2	ATS_RUN_LEVEL
3	Reserved
4	ATS_CT0_DEFAULT_STATE
5	ATS_CT1_DEFAULT_STATE
6	ATS_CT2_DEFAULT_STATE
7	ATS_CT3_DEFAULT_STATE
8	ATS_IMMDT_JUMP_SEL
9	ATS_IMMDT_JUMP_SEL
10	ATS_IMMDT_JUMP_SEL
11	ATS_IMMDT_JUMP_SEL
12	ATS_IMMDT_JUMP_COND
13	ATS_IMMDT_JUMP_EN
14	Reserved
15	Reserved
16	ATS_IDX_JUMP
17	ATS_IDX_JUMP
18	ATS_IDX_JUMP
19	ATS_IDX_JUMP
20	ATS_IDX_JUMP
21	ATS_IDX_JUMP
22	ATS_IDX_JUMP
23	ATS_IDX_JUMP
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	ATS_TRIG_SEL
29	ATS_TRIG_SEL
30	ATS_TRIG_SEL
31	ATS_TRIG_SEL

ATS_RUN_LEVEL R/W, ATS_CONTROL[2..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

These bits control the operation of the ATS. These bits are used to start and stop the sequencer. They can also be used to program the table to jump to a new section. Jumps are always synchronous (i.e. not immediate). Jumps will only occur from an index in the sequence that has the ATS_END_OF_SEQUENCE bit set.

This bit can be read at any time in order get the current status.

The following table shows the command available for this register.

ATS_RUN_LEVEL	Meaning
0 (000b)	Idle - ATS is not running
1 (001b)	Run - Start running immediately from index in the ATS_IDX_JUMP register
2 (010b)	Jump - Jump to index set in the ATS_IDX_JUMP register next time the current index has the ATS_END_OF_SEQUENCE bit set to 1
3 (011b)	Stop - Stop running the next time the current index has the ATS_END_OF_SEQUENCE bit set to 1
4 (100b)	Abort - Stop running immediately

ATS_CT0_DEFAULT_STATE R/W, ATS_CONTROL[4], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This is the output state of CT0 when the ATS is Idle.

ATS_CT1_DEFAULT_STATE R/W, ATS_CONTROL[5], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This is the output state of CT1 when the ATS is Idle.

ATS_CT2_DEFAULT_STATE R/W, ATS_CONTROL[6], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This is the output state of CT2 when the ATS is Idle.

ATS_CT3_DEFAULT_STATE R/W, ATS_CONTROL[7], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This is the output state of CT3 when the ATS is Idle.

ATS_IMMDT_JUMP_SEL R/W, ATS_CONTROL[8..11], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

These bits select the source of the Immediate Jump function.

ATS_IMMDT_JUMP_SEL	Meaning
0 (0000b)	Selected trigger (VGFx_TRIG_SEL)
1 (0001b)	Selected encoder A (VFGx_ENCA_SEL)
2 (0010b)	Selected encoder B (VFGx_ENCB_SEL)
3 (0011b)	Selected quad encoder output (VFGx_ENCQ_SEL)
4 (0100b)	Gated trigger (VGFx_TRIG_SEL gated by VFGx_ENCB_SEL)
5 (0101b)	Selected encoder divider output (VFGx_ENCDIV_SEL)
6 (0110b)	AE start of frame (Y Window Opens)
7 (0111b)	AE end of frame (Y Window Closes)
8 (1000b)	AE start of line (X Window Opens)
9 (1001b)	AE end of line (X Window Closes)
10 (1010b)	FVAL rising edge*
11 (1011b)	FVAL falling edge*
12 (1100b)	LVAL rising edge*
13 (1101b)	LVAL falling edge*
14 (1110b)	Reserved
15 (1111b)	Reserved

*Note: * FVAL and LVAL triggers are only support on the Axion models*

ATS_IMMDT_JUMP_COND R/W, ATS_CONTROL[12], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This controls the polarity of the Immediate Jump function.

ATS_IMMDT_JUMP_COND	Meaning
0	Rising edge
1	Falling edge

ATS_IMMDT_JUMP_EN R/W, ATS_CONTROL[12], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Set this bit to 1 to enable Immediate Jump function.

ATS_IDX_JUMP R/W, ATS_CONTROL[23..16], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This is the entry that the table will start from when the ATS_RUN_LEVEL register is set to Run.

This is also the entry that the table will jump to (synchronously) when the ATS_RUN_LEVEL register is set to Jump.

ATS_TRIG_SEL R/W, ATS_CONTROL[28..31], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

These bits select the source of the ATS trigger.

TS_TRIG_SEL	Meaning
0 (0000b)	Selected trigger (VGFx_TRIG_SEL)
1 (0001b)	Selected encoder A (VFGx_ENCA_SEL)
2 (0010b)	Selected encoder B (VFGx_ENCB_SEL)
3 (0011b)	Selected quad encoder output (VFGx_ENCQ_SEL)
4 (0100b)	Gated trigger (VGFx_TRIG_SEL gated by VFGx_ENCB_SEL)
5 (0101b)	Selected encoder divider output (VFGx_ENCDIV_SEL)
6 (0110b)	Acquisition start of frame (Y Window Open)
7 (0111b)	Acquisition end of frame (Y Window Close)
8 (1000b)	Acquisition start of line (X Window Open)
9 (1001b)	Acquisition end of line (X Window Open)
10 (1010b)	FVAL rising edge*
11 (1011b)	FVAL falling edge*
12 (1100b)	LVAL rising edge*
13 (1101b)	LVAL falling edge*
14 (1110b)	Reserved
15 (1111b)	Reserved

*Note: * FVAL and LVAL triggers are only support on the Axion models*

4.6 ATS_TABLE_CONTROL

Bit	Name
0	ATS_IDX_ACCESS
1	ATS_IDX_ACCESS
2	ATS_IDX_ACCESS
3	ATS_IDX_ACCESS
4	ATS_IDX_ACCESS
5	ATS_IDX_ACCESS
6	ATS_IDX_ACCESS
7	ATS_IDX_ACCESS
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

ATS_IDX_ACCESS R/W, ATS_TABLE_CONTROL[7..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Indirect access to the TS table. Set this bitfield to the index value that you wish to modify. Access is done through via TS_TABLE_ENTRY register.

4.7 ATS_TABLE_ENTRY

Bit	Name
0	ATS_NEXT
1	ATS_NEXT
2	ATS_NEXT
3	ATS_NEXT
4	ATS_NEXT
5	ATS_NEXT
6	ATS_NEXT
7	ATS_NEXT
8	Reserved
9	Reserved
10	ATS_RESOLUTION
11	ATS_RESOLUTION
12	ATS_STATE_CT0
13	ATS_STATE_CT1
14	ATS_STATE_CT2
15	ATS_STATE_CT3
16	ATS_TRIG_SEL_SEL
17	ATS_COUNT
18	ATS_COUNT
19	ATS_COUNT
20	ATS_COUNT
21	ATS_COUNT
22	ATS_COUNT
23	ATS_COUNT
24	ATS_COUNT
25	ATS_COUNT
26	ATS_COUNT
27	ATS_CONDITION
28	ATS_CONDITION
29	ATS_CONDITION
30	ATS_TERMINATE
31	ATS_END_OF_SEQUENCE

ATS_NEXT R/W, ATS_TABLE_ENTRY[7..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Index of next pulse. Only follow if ATS_TERMINATE = 0.

ATS_ RESOLUTION R/W, ATS_TABLE_ENTRY[11..10], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The time units of this pulse. The length of this pulse is set in the register ATS_COUNT. The following table shows the available resolutions.

ATS_RESOLUTION	Meaning
0 (000b)	100 nanoseconds
1 (001b)	100 microseconds
2 (010b)	100 milliseconds
3 (011b)	100 seconds

ATS_STATE_CT0 R/W, ATS_TABLE_ENTRY[12], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The level of the CT0 signal for this pulse.

ATS_STATE_CT1 R/W, ATS_TABLE_ENTRY[13], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The level of the CT1 signal for this pulse.

ATS_STATE_CT2 R/W, ATS_TABLE_ENTRY[14], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The level of the CT2 signal for this pulse.

ATS_STATE_CT3 R/W, ATS_TABLE_ENTRY[15], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The level of the CT3 signal for this pulse.

ATS_TRIG_SEL_SEL R/W, ATS_TABLE_ENTRY[16], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

If this pulse is triggered, this bit sets which trigger will be used.

ATS_TRIG_SEL_SEL	Meaning
0	The trigger selected by ATS_TRIG_SEL_0
1	The trigger selected by ATS_TRIG_SEL_1

ATS_COUNT R/W, ATS_TABLE_ENTRY[26..17], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The length of this pulse. The units for the length are set in the ATS_RESOLUTION register.

ATS_CONDITION R/W, ATS_TABLE_ENTRY[29..27], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register is used to control the conditions under which this pulse will be output. The following table shows the options for this bitfield.

ATS_CONDITION	Condition when pulse is output
0 (000b)	Immediate
1 (001b)	Rising edge of trigger
2 (010b)	Falling edge of trigger
3 (011b)	Trigger high
4 (100b)	Trigger low
5 (101b)	Both rising and falling edge of trigger

ATS_TERMINATE R/W, ATS_TABLE_ENTRY[30], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

When this bit is set to 0, the table will stop running after the current pulse is output. The ATS_RUN_LEVEL bitfield will then read back idle.

When this bit is set to 1, the table will jump to the index set in the ATS_NEXT bitfield after the current pulse is finished.

ATS_END_OF_SEQUENCE R/W, ATS_TABLE_ENTRY[31], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

If this bit is set to 1 and ATS_RUN_LEVEL is set to Jump, the ATS will jump to the index set in the ATS_INDX_JUMP bitfield after the current pulse is output. This bit allows for synchronous switching between one section of the table and another section.

If the ATS_RUN_LEVEL bitfield is not set to Jump, then this bitfield will have no effect.

If this bit is set to 0, the ATS will not jump from this index.

The Aon, Axion, Claxon and Cyton I/O System

Chapter 5

5.1 Introduction

The I/O system on the Aon, Axion, Claxon and Cyton families of frame grabbers are based on the Karbon family with some minor changes. This system provides unprecedented flexibility. The goal of this system is to handle all the I/O needs of any machine vision application connected to the real world in a wide variety of ways. The goal is flexibility and observability.

5.1.1 Concepts

The basic concept is that the outside world of a machine vision system can have a wide variety of signals, possibly using different electrical standards. The Cyton and/or Axion user can choose whichever ones best suit their needs. These inputs can then be routed to a wide range of internal destinations. Also, the board can generate its own signals of use in this system.

Once the input sources are chosen, they are routed to one of a number of internal signals. These internal signals can then be used to control a wide variety of functions. For example, a function might be to cause the board to acquire a frame.

Finally the internal signals can be routed off the board to a wide number of destinations. These can be used to control something in the outside world. For example, a signal might be routed such that it fires a strobe light or initiates the start of exposure in a camera.

The state of all of the possible inputs can be observed by software at any time by peeking the associated RD_XXX bit.

5.1.2 I/O Between Virtual Frame Grabbers

Because BitFlow's frame grabber can acquire from more than one camera, it has always been a contention between the desire to let each camera be independent, for example, each with its own trigger, and for them to be synchronized, i.e. all cameras using one trigger. The Cyton and Axion I/O system provides the best of both worlds by fully supporting independent and synchronized triggers using the same flexibility as each individual VFG gets.

What this means is that one of the sources, the master VFG's trigger can be the trigger for all of the VFGs. Thus whatever signal is triggering VFG0, can also trigger all the other VFGs on the board. Of course, each VFG can choose to use the master VFG's trigger, or choose amongst its own sources. Further, the triggers can be synchronized while the encoders are independent.

5.2 Overview of the I/O System Routing

Figure 5-1 below shows a generic version of the I/O system routing. For each internal signal (A and B in this case) a source must be chosen. Each internal signal can be used to control one or more internal circuits or can be routed straight to an output signal. For each output signal (X & Y in this case) a source must be chosen. In addition, there are internal signal generators that can be chosen as a source for an internal signal or an external signal. Even the internal signal generators can be triggered by an internal signal.

Note: Figure 5-1 is a schematic version of the actual circuit, it is greatly simplified to make it easier to understand.

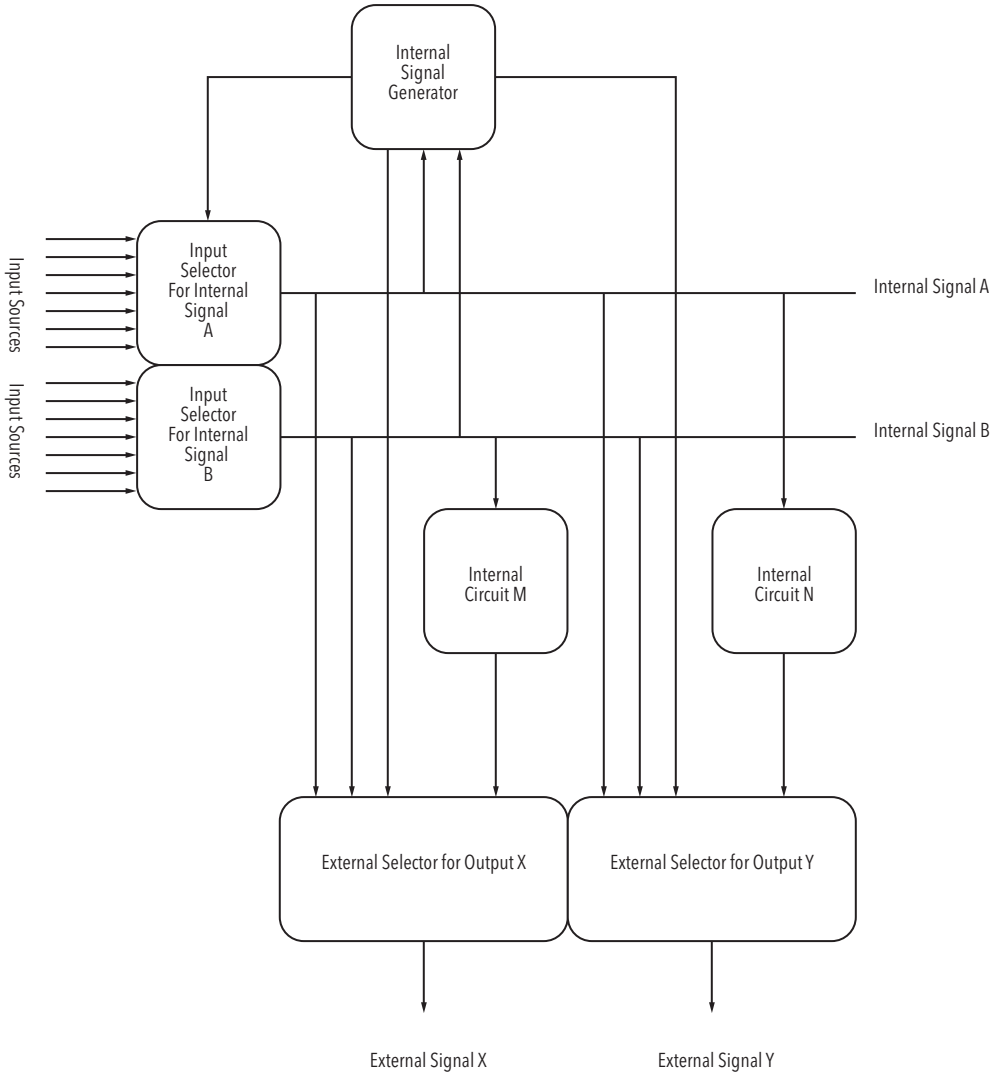

Figure 5-1 Conceptual I/O System Routing

5.3 Input Selection

Figure 5-2 illustrates the I/O System input selection circuitry. As discussed above, each internal signal has its own selector. For each internal signal there are 64 possible sources.

Note: The signals BOX_IN_XXX are available via an external I/O Box, the BitFlow BitBox, which can be mounted on an external rail system. Contact BitFlow for more information on the BitBox.

Note: Each VFG has a copy of the circuit shown in Figure 5-2. This is why the outputs do not specify the VFG number (e.g. "VFGx_TRIG_SEL). However, some inputs do specify the VFG number (e.g. VFG0_TRIG_SEL), which means this input comes explicitly from VFG0.

Note: The Aon-CXP, Claxon-CXP1, Axion-1xE and Axion-1xB have only one VFG.

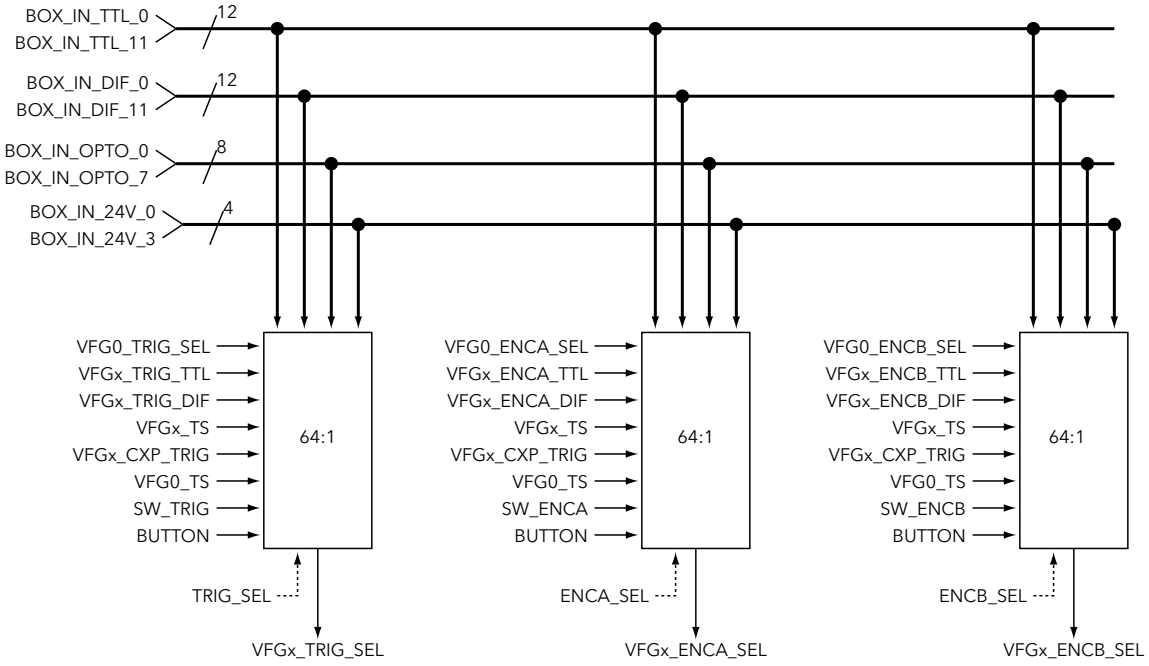

Figure 5-2 I/O System Input Selection

5.4 Internal Signals

There are five internal I/O signals. The "x" in the signal name refers to the VFG number of the current VFG being used. In general, the "x" is always used as all VFGs are symmetrical. The one exceptions is VFG0, which can route many of its signals to other VFGs on the same physical board. For example VFG2 can use the trigger selected on VFG 0 (i.e. VFG0_TRIG_SEL) as its trigger source.

- VFGx_TRIG_SEL - normally used as a frame trigger, can also be used to initiate acquisition of N frames, or can be used to control the start and end of a frame when used with a line scan camera.
- VFGx_ENCA_SEL - Normally used as a line trigger, to initiate the capture of one line. When using a single phase encoder, this is the signal to use.
- VFGx_ENCB_SEL - Normally used as a line trigger when using a quadrature encoder. The encoder B should be the second phase of the quadrature encoder with Encoder A being the first phase.
- VFGx_ENCDIV_SEL - This signal is the output of the encoder divider (multiplier) circuit. This circuit can be driven by more than one source. The output signal is related to the input signal, but the frequency is either divided down or multiplied up.
- VFGx_ENCQ_SEL - This signal is the output of the quadrature encoder circuit. This circuit takes two inputs, the selected encoder A and the selected encoder B. The output signal follows the rules as programmed in to the quadrature encoder circuit, see Section 8.1 for more information.

Note: The internal signals have names such as "Trigger" and "Encoder" because they are hardwired to certain internal circuits. However, if you are not using them for this functionality, they can be used for any purpose that the routing supports.

Each of the internal signals is hardwired to a number of destinations. Even though a signal might be connected to a circuit, the circuit may not be "listening" to the signal. Each circuit has its own control registers which tells it to use a given signal or not. Figure 5-3 shows all the internal circuits that each internal signal is connected to.

Figure 5-4 shows the details of the encoder handler block that is part of Figure 5-3.

The Filter is used to remove unwanted noise on the incoming signal. The filter is programmable and it will "swallow" a pulse shorter than the programmed size.

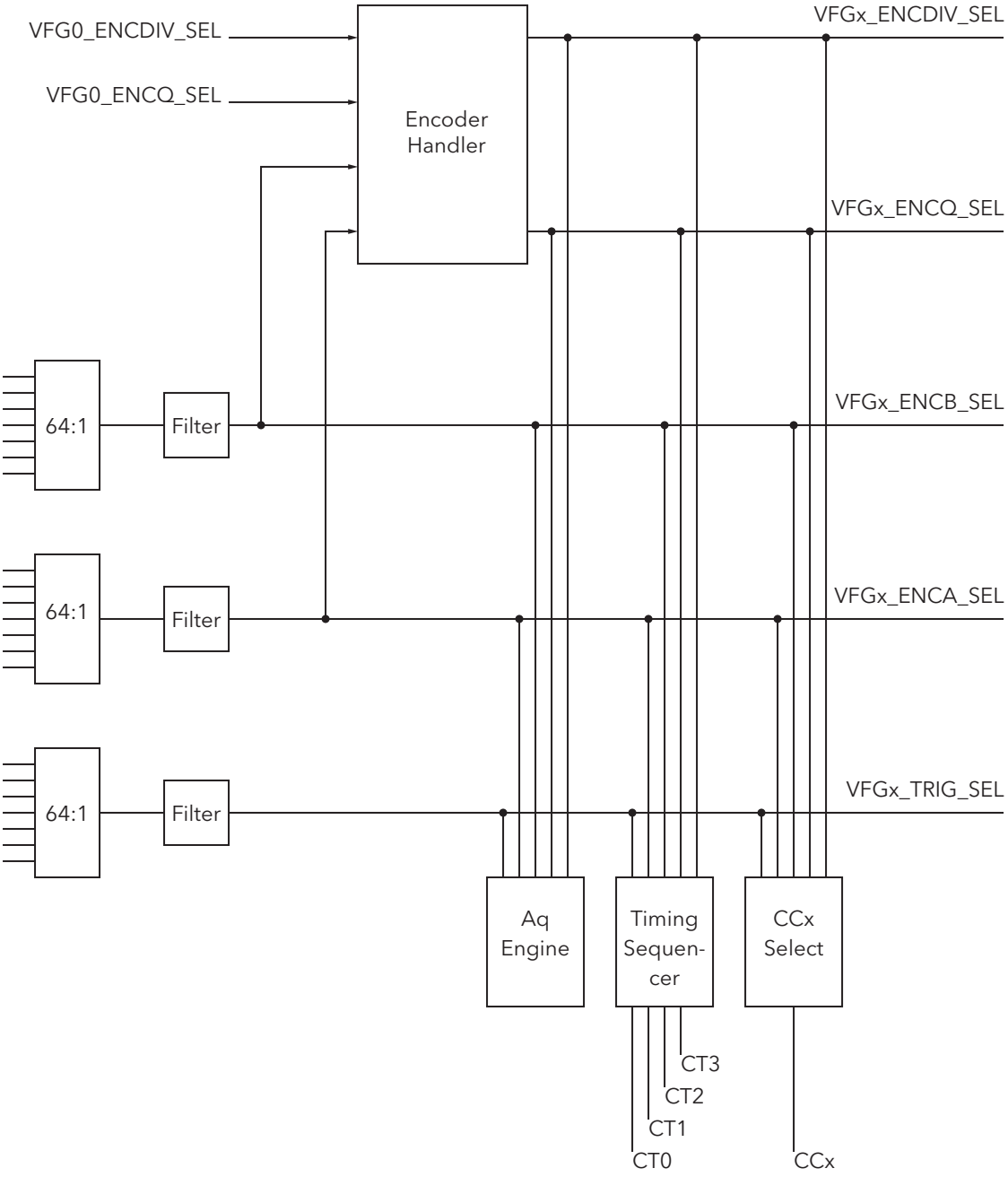

Figure 5-3 Internal Signal Routing

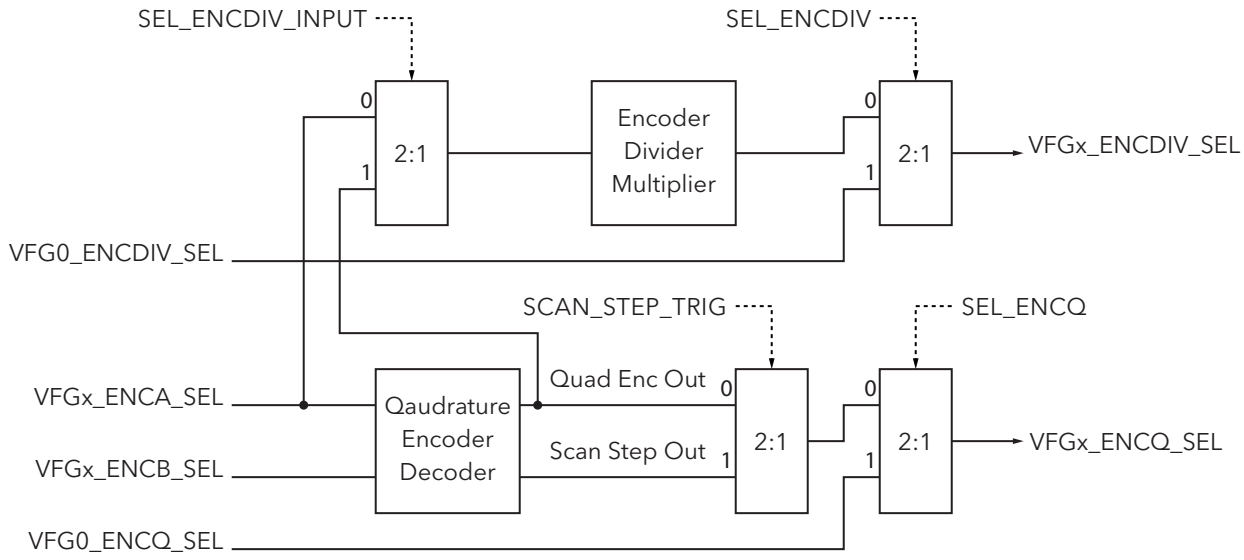

Figure 5-4 Encoder Handler

5.5 Output Signal Selection

There are four dynamic output signals for each VFG: CC1, CC2, CC3 and CC4. These dynamic signals can be driven by a variety of sources as shown in Figure 5-5.

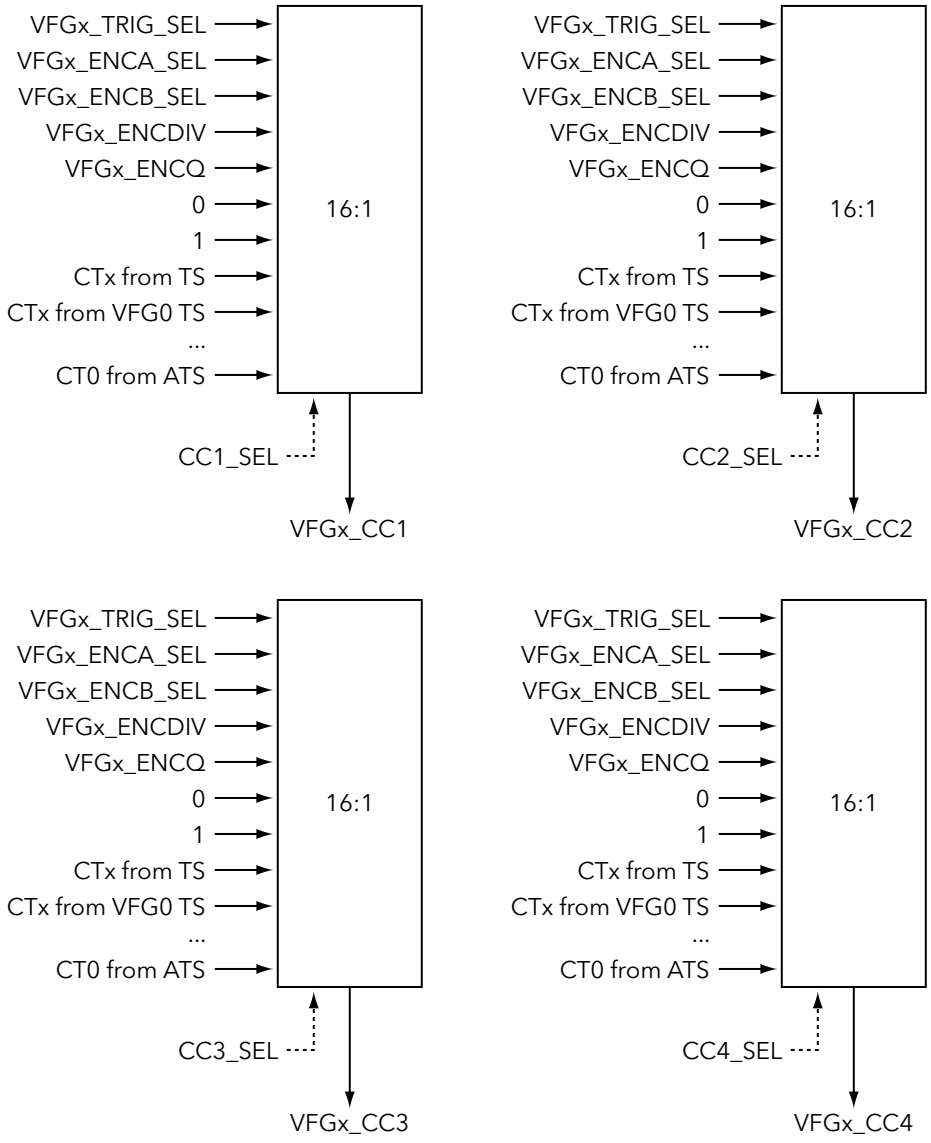

Figure 5-5 Output Signal Source Selection

5.6 I/O Connector Output Signal Routing

The CCx output signals are routed to the CXP link or the board's main I/O connector Figure 5-6 illustrates how they are routed.

Note: Each VFG has an instance of the circuit shown below. This means each VFG has its own CC2, CC3 and CC4 signals present on the board's main I/O connector.

Note: The signals VFGx_CC1 to VFGx_CC3 can be sourced from many different signals. Please see Section 5.5 for more information on selecting the source.

Note: The signals VFGx_CC1 to VFGx_CC3 can also be routed to the BitBox, which is an externally mountable I/O Box.

Figure 5-6 Output Signal Routing

5.7 BitBox Output Signal Routing

The BitFlow BitBox has 3 banks of 12 outputs. One bank is TTL, one bank is differential and one bank is a mix of Optocoupled and Open Collector outputs. Each of the 36 outputs can be set to a static output level, or controlled by a dynamic source (waveform). Figure 5-7 shows how the outputs are controlled.

Note: Figure 5-7 shows the choices and routing for just 3 outputs. A total of 12 of these circuits (each with their own control registers) are provided on the board in order to control 36 outputs.

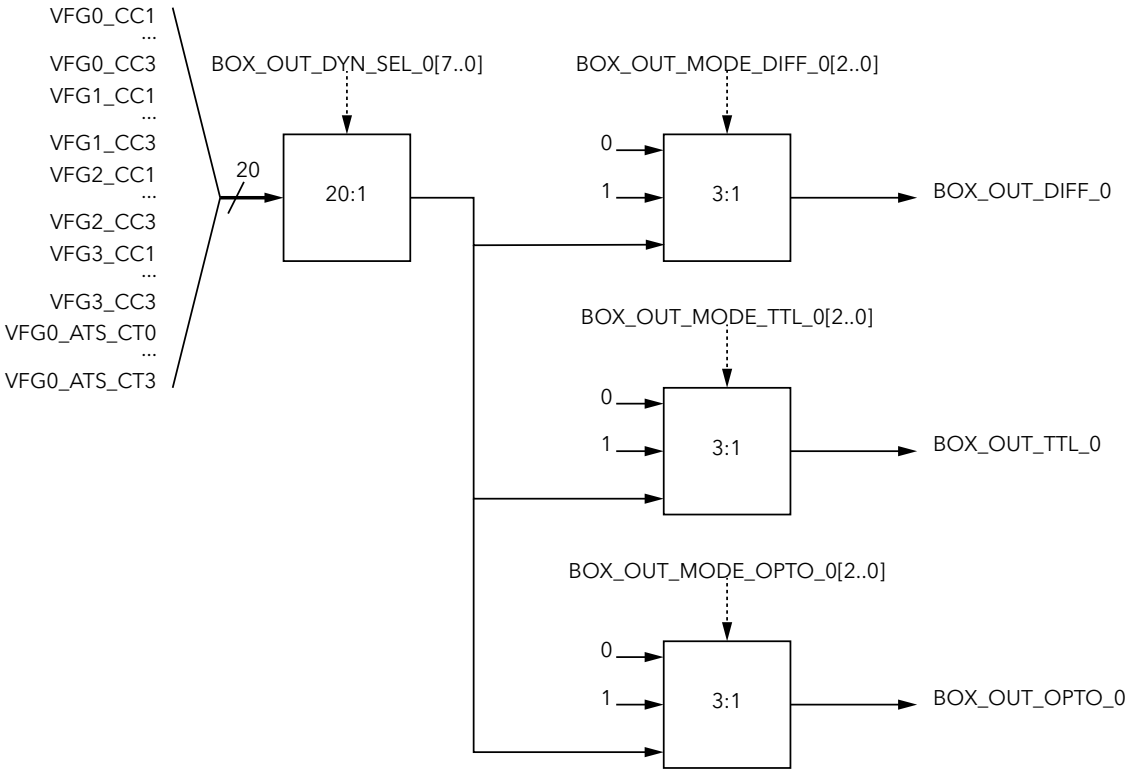

Figure 5-7 BitBox Output Signal Routing

Note: The BitFlow BitBox is an externally rail mounted box which can take a wide variety of inputs and outputs. It is connected to the Cyton/Axion though a small cable. Contact BitFlow for more information on the BitBox.

The I/O System Registers

Chapter 6

6.1 Introduction

The registers documented in this section are used to control the I/O system on the Aon, Axion, Claxon and Cyton.

6.2 CON60

Bit	Name
0	RD_BOX_IN_TTL_0
1	RD_BOX_IN_TTL_1
2	RD_BOX_IN_TTL_2
3	RD_BOX_IN_TTL_3
4	RD_BOX_IN_TTL_4
5	RD_BOX_IN_TTL_5
6	RD_BOX_IN_TTL_6
7	RD_BOX_IN_TTL_7
8	RD_BOX_IN_TTL_8
9	RD_BOX_IN_TTL_9
10	RD_BOX_IN_TTL_10
11	RD_BOX_IN_TTL_11
12	RD_BOX_IN_DIF_0
13	RD_BOX_IN_DIF_1
14	RD_BOX_IN_DIF_2
15	RD_BOX_IN_DIF_3
16	RD_BOX_IN_DIF_4
17	RD_BOX_IN_DIF_5
18	RD_BOX_IN_DIF_6
19	RD_BOX_IN_DIF_7
20	RD_BOX_IN_DIF_8
21	RD_BOX_IN_DIF_9
22	RD_BOX_IN_DIF_10
23	RD_BOX_IN_DIF_11
24	ENINT_CXP
25	INT_CXP
26	Reserved
27	Reserved
28	Reserved
29	SW_TRIG
30	SW_ENCA
31	SW_ENCB

RD_BOX_IN_TTL_X RO, CON60[11..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

These bits reflect the real-time state of the 12 TTL inputs on the BitBox.

RD_BOX_IN_DIF_X RO, CON60[23..12], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

These bits reflect the real-time state of the 12 differential inputs on the BitBox.

ENINT_CXP R/W, CON60[24], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit enables interrupts from the CXP subsystem.

INT_CXP RO, CON60[25], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit indicates the existence of an interrupt from the CXP subsystem. The individual interrupt must be cleared in the CXP subsystem in order for this bit to reset.

SW_TRIG R/W, CON60[29], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Writing the bit to 1 causes the internal software trigger signal to be asserted. Writing it to a 0 will de-assert the internal software trigger signal.

SW_ENCA R/W, CON60[30], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Writing the bit to 1 causes the internal encoder A signal to be asserted. Writing it to a 0 will de-assert the internal encoder A signal.

SW_ENCB R/W, CON60[31], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Writing the bit to 1 causes the internal encoder B signal to be asserted. Writing it to a 0 will de-assert the internal encoder B signal.

6.3 CON61

Bit	Name
0	RD_BOX_IN_OPTO_0
1	RD_BOX_IN_OPTO_1
2	RD_BOX_IN_OPTO_2
3	RD_BOX_IN_OPTO_3
4	RD_BOX_IN_OPTO_4
5	RD_BOX_IN_OPTO_5
6	RD_BOX_IN_OPTO_6
7	RD_BOX_IN_OPTO_7
8	RD_BOX_IN_24V_0
9	RD_BOX_IN_24V_1
10	RD_BOX_IN_24V_2
11	RD_BOX_IN_24V_3
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	RD_CXP_TRIG_OUT
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

RD_BOX_IN_OPTO_X RO, CON61[7..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

These bits reflect the real-time state of the eight Opto-Isolated inputs on the BitBox.

RD_BOX_IN_24V_X RO, CON61[11..8], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

These bits reflect the real-time state of the four 24V inputs on the BitBox.

RD_CXP_TRIG_OUT RO, CON61[23], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit reflects the real-time state of the CXP trigger signal going to the camera.

6.4 CON62

Bit	Name
0	RD_TRIG_TTL
1	RD_TRIG_DIF
2	RD_TRIG_VFG0
3	RD_SCAN_STEP
4	RD_SW_TRIG
5	RD_ENCA_TTL
6	RD_ENCA_DIF
7	RD_ENCA_VFG0
8	RD_ENCA_SW
9	RD_ENCB_TTL
10	RD_ENCB_DIF
11	RD_ENCB_VFG0
12	RD_ENCB_SW
13	RD_BUTTON
14	Reserved
15	DIV_RESET_DISABLE
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	RD_CXP_TRIG_IN
25	EN_TRIG
26	EN_ENCA
27	EN_ENCB
28	Reserved
29	RD_ENCB_SELECTED
30	RD_ENCA_SELECTED
31	RD_TRIG_SELECTED

RD_TRIG_TTL RO, CON62[0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit reflects the real-time state of the VFG's TTL trigger input.

RD_TRIG_DIF RO, CON62[1], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit reflects the real-time state of the VFG's differential trigger input.

RD_TRIG_VFG0 RO, CON62[2], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit reflects the real-time state of VFG0's selected trigger signal.

RD_SCAN_STEP RO, CON62[3], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit reflects the real-time state of the VFG's scan step circuitry output (from the quadrature encoder circuit).

RD_SW_TRIG RO, CON62[4], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit reflects the real-time state of the VFG's software trigger.

RD_ENCA_TTL RO, CON62[5], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit reflects the real-time state of the VFG's TTL encoder A input.

RD_ENCA_DIF RO, CON62[6], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit reflects the real-time state of the VFG's differential encoder A input.

RD_ENCA_VFG0 RO, CON62[7], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit reflects the real-time state of VFG0's selected encoder A signal.

RD_ENCA_SW RO, CON62[8], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit reflects the real-time state of the VFG's software encoder A.

RD_ENCB_TTL RO, CON62[9], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit reflects the real-time state of the VFG's TTL encoder B input.

RD_ENCB_DIF RO, CON62[10], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit reflects the real-time state of the VFG's differential encoder B input.

RD_ENCB_VFG0 RO, CON62[11], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit reflects the real-time state of VFG0's selected encoder B signal.

RD_ENCB_SW RO, CON62[12], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit reflects the real-time state of the VFG's software encoder B.

RD_BUTTON RO, CON62[13], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit reflects the real-time state of the VFG's button input.

DIV_RESET_DISABLE R/W, CON62[15], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit disables the Divider/Multiplier auto-reset function.

RD_CXP_TRIG_IN RO, CON62[24], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit reflects the real-time state of the CXP trigger signal coming from the camera.

EN_TRIG R/W, CON62[25], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit is used to enable the selected trigger.

EN_ENCA R/W, CON62[26], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit is used to enable the selected encoder A.

EN_ENCB R/W, CON62[27], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit is used to enable the selected encoder B.

RD_ENCB_SELECTED RO, CON62[29], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The bit reflects the real-time status of the VFG's select encoder B input.

RD_ENCA_SELECTED RO, CON62[30], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The bit reflects the real-time status of the VFG's selected encoder A input.

RD_TRIG_SELECTED RO, CON62[31], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The bit reflects the real-time status of the VFG's selected trigger input.

6.5 CON63

Bit	Name
0	SEL_TRIG
1	SEL_TRIG
2	SEL_TRIG
3	SEL_TRIG
4	SEL_TRIG
5	SEL_TRIG
6	SEL_ENCA
7	SEL_ENCA
8	SEL_ENCA
9	SEL_ENCA
10	SEL_ENCA
11	SEL_ENCA
12	SEL_ENCB
13	SEL_ENCB
14	SEL_ENCB
15	SEL_ENCB
16	SEL_ENCB
17	SEL_ENCB
18	SEL_CC1
19	SEL_CC1
20	SEL_CC1
21	SEL_CC1
22	SEL_CC2
23	SEL_CC2
24	SEL_CC2
25	SEL_CC2
26	Reserved
27	Reserved
28	SEL_LED
29	SEL_LED
30	SEL_LED
31	SEL_LED

SEL_TRIG R/W, CON63[5..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Selects the VFG's trigger source.

SEL_TRIG	Source
0 (000000b)	Forced low
1 (000001b)	Forced high
2 (000010b)	This VFG's differential trigger VFGx_TRIGGER+/-
3 (000011b)	This VFG's TTL trigger VFGx_TRIGGER_TTL
4 (000100b)	Selected trigger from VFG0
5 (000101b)	This VFG's Timing Sequencer (TS)
6 (000110b)	Button
7 (000111b)	The camera's CXP trigger
8 (001000b)	This VFG's software trigger, SW_TRIG
9 (001001b)	This VFG's scan step circuit
10 (001010b)	VFG0's Timing Sequencer (TS)
11-27	Reserved
28 to 39	BOX_IN_TTL_0 to BOX_IN_TTL_11
40 to 51	BOX_IN_DIF_0 to BOX_IN_DIF_11
52 to 59	BOX_IN_OPTO_0 to BOX_IN_OPTO_7
61 to 63	BOX_IN_24V_0 to BOX_IN_24V_3

SEL_ENCA R/W, CON63[11..6], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Selects this VFG's encoder A source.

SEL_ENCA	Source
0 (000000b)	Forced low
1 (000001b)	Forced high
2 (000010b)	This VFG's differential encoder A, VFGx_ENCA+/-
3 (000011b)	This VFG's TTL encoder A, VFGx_ENCA_TTL
4 (000100b)	Selected encoder A from VFG0
5 (000101b)	This VFG's Timing Sequencer (TS)
6 (000110b)	Button
7 (000111b)	The camera's CXP trigger

SEL_ENCA	Source
8 (001000b)	This VFG's software encoder A, SW_ENCA
9 (001001b)	VFG0's Timing Sequencer (TSS
10-27	Reserved
28 to 39	BOX_IN_TTL_0 to BOX_IN_TTL_11
40 to 51	BOX_IN_DIF_0 to BOX_IN_DIF_11
52 to 59	BOX_IN_OPTO_0 to BOX_IN_OPTO_7
61 to 63	BOX_IN_24V_0 to BOX_IN_24V_3

SEL_ENCB R/W, CON63[17..12], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Selects the source of encoder B.

SEL_ENCB	Source
0 (000000b)	Forced low
1 (000001b)	Forced high
2 (000010b)	This VFG's differential encoder B VFGx_ENCB+/-
3 (000011b)	This VFG's TTL encoder B VFGx_ENCB_TTL
4 (000100b)	Selected encoder B from VFG0
5 (000101b)	This VFG's Timing Sequencer (TS)
6 (000110b)	Button
7 (000111b)	The camera's CXP trigger
8 (001000b)	This VFG's software encoder B, SW_ENCB
9 (001001b)	VFG0's Timing Sequencer (TS)
10-27	Reserved
28 to 39	BOX_IN_TTL_0 to BOX_IN_TTL_11
40 to 51	BOX_IN_DIF_0 to BOX_IN_DIF_11
52 to 59	BOX_IN_OPTO_0 to BOX_IN_OPTO_7
61 to 63	BOX_IN_24V_0 to BOX_IN_24V_3

SEL_CC1 R/W, CON63[21..18], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Selects the source of CC1.

SEL_CC1	Source
0 (0000b)	Forced low
1 (0001b)	Forced high
2 (0010b)	CT0 (from TS)
3 (0011b)	CT1 (from TS)
4 (0100b)	CT2 (from TS)
5 (0101b)	CT3 (from TS)
6 (0110b)	VFGx_TRIG_SEL
7 (0111b)	VFGx_ENCA_SEL
8 (1000b)	VFGx_ENCB_SEL
9 (1001b)	VFG0_CT0
10 (1010b)	VFG0_CT1
11 (1011b)	VFG0_CT2
12 (1100b)	VFG0_CT3
13 (1101b)	VFGx_ENCDIV_SEL
14 (1110b)	VFGx_ENCQ_SEL
15 (111b)	CT0 (from ATS)

SEL_CC2 R/W, CON63[25..22], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Selects the source of CC2.

SEL_CC2	Source
0 (0000b)	Forced low
1 (0001b)	Forced high
2 (0010b)	CT0 (from TS)
3 (0011b)	CT1 (from TS)
4 (0100b)	CT2 (from S)
5 (0101b)	CT3 (from TS)

SEL_CC2	Source
6 (0110b)	VFGx_TRIG_SEL
7 (0111b)	VFGx_ENCA_SEL
8 (1000b)	VFGx_ENCB_SEL
9 (1001b)	VFG0_CT0
10 (1010b)	VFG0_CT1
11 (1011b)	VFG0_CT2
12 (1100b)	VFG0_CT3
13 (1101b)	VFGx_ENCDIV_SEL
14 (1110b)	VFGx_ENCQ_SEL
15 (111b)	CT0 (from ATS)

SEL_LED R/W, CON63[31..28], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Selects the source of the VFG's green LED. The LED receives a 1/2 second pulse every time the selected event occurs.

SEL_LED	Source
0 (0000b)	Board emits an interrupt to the host
1 (0001b)	VFGx_TRIG_SEL
2 (0010b)	VFG0_TRIG_SEL
3 (0011b)	Button
4 (0100b)	FVAL from camera
5 (0101b)	VAW
6 (0110b)	VWIN
7 (0111b)	CC1
8 (1000b)	CC2
9 (1001b)	CC3
10 (1010b)	CC4
11 (1011b)	VFGx_NTG or VFGx_TS

SEL_LED	Source
12 (1100b)	VFG0_NTG or VFG0_TS
13 (1101b)	AQSTAT[1]
14 (1110b)	Overstep, OVS
15 (1111b)	Reserved

6.6 CON64

Bit	Name
0	SEL_CC3
1	SEL_CC3
2	SEL_CC3
3	SEL_CC3
4	SEL_CC4
5	SEL_CC4
6	SEL_CC4
7	SEL_CC4
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	TRIGPOL
14	ENCA_POL
15	ENCB_POL
16	RD_CC1
17	RD_CC2
18	RD_CC3
19	RD_CC4
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	UID
28	LED_RED
29	LED_ORANGE
30	LED_GREEN
31	LED_BLUE

SEL_CC3 R/W, CON64[3..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Selects the source of CC3.

SEL_CC3	Source
0 (0000b)	Forced low
1 (0001b)	Forced high
2 (0010b)	CT0 (from TS)
3 (0011b)	CT1 (from TS)
4 (0100b)	CT2 (from TS)
5 (0101b)	CT3 (from TS)
6 (0110b)	VFGx_TRIG_SEL
7 (0111b)	VFGx_ENCA_SEL
8 (1000b)	VFGx_ENCB_SEL
9 (1001b)	VFG0_CT0
10 (1010b)	VFG0_CT1
11 (1011b)	VFG0_CT2
12 (1100b)	VFG0_CT3
13 (1101b)	VFGx_ENCDIV_SEL
14 (1110b)	VFGx_ENCQ_SEL
15 (111b)	CT0 (from ATS)

SEL_CC4 R/W, CON64[7..4], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Selects the source of CC4.

SEL_CC4	Source
0 (0000b)	Forced low
1 (0001b)	Forced high
2 (0010b)	CT0 (from TS)
3 (0011b)	CT1 (from TS)
4 (0100b)	CT2 (from TS)
5 (0101b)	CT3 (from TS)
6 (0110b)	VFGx_TRIG_SEL
7 (0111b)	VFGx_ENCA_SEL

SEL_CC4	Source
8 (1000b)	VFGx_ENCB_SEL
9 (1001b)	VFG0_CT0
10 (1010b)	VFG0_CT1
11 (1011b)	VFG0_CT2
12 (1100b)	VFG0_CT3
13 (1101b)	VFGx_ENCDIV_SEL
14 (1110b)	VFGx_ENCQ_SEL
15 (111b)	CT0 (from ATS)

TRIGPOL R/W, CON64[13], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Selects the edge of the trigger signal that corresponds to its assertion.

TRIGPOL	Meaning
0	Trigger asserted on rising edge
1	Trigger asserted on falling edge

ENCA_POL R/W, CON64[14], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Selects the edge of encoder A signal that corresponds to its assertion.

ENCA_POL	Meaning
0	Encoder A asserted on rising edge
1	Encoder A asserted on falling edge

ENCB_POL R/W, CON64[15], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Selects the edge of encoder B signal that corresponds to its assertion.

ENCB_POL	Meaning
0	Encoder B asserted on rising edge
1	Encoder B asserted on falling edge

RD_CC1 RO, CON64[16], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Returns the current state of the CC1 output.

RD_CC1	Meaning
0	Output is low
1	Output is high

RD_CC2 RO, CON64[17], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Returns the current state of the CC2 output.

RD_CC2	Meaning
0	Output is low
1	Output is high

RD_CC3 RO, CON64[18], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Returns the current state of the CC3 output.

RD_CC3	Meaning
0	Output is low
1	Output is high

RD_CC4 RO, CON64[19], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Returns the current state of the CC4 output.

RD_CC4	Meaning
0	Output is low
1	Output is high

UID R/W, CON64[27], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Setting this bit to 1enables Unit ID mode. This mode causes the LEDs on the back and top of the board to blink RED. This mode can be used to identify which physical board is which VFG. Setting this bit to 0 disables UID mode.

LED_RED R/W, CON64[28], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Setting this bit to 1 turns the red LED on. Setting this bit to 1 on any VFG turns the LED on. This bit must be set to 0 on all VFGs in order to turn this LED off.

LED_ORANGE R/W, CON64[29], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Setting this bit to 1 turns the orange LED on. Setting this bit to 1 on any VFG turns the LED on. This bit must be set to 0 on all VFGs in order to turn this LED off.

LED_GREEN R/W, CON64[30], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Setting this bit to 1 turns the green LED on. Setting this bit to 1 on any VFG turns the LED on. This bit must be set to 0 on all VFGs in order to turn this LED off.

6.7 TRIG_OPTS

Bit	Name
0	TRIG_FILTER
1	TRIG_FILTER
2	TRIG_FILTER
3	TRIG_FILTER
4	TRIG_FILTER
5	TRIG_FILTER
6	TRIG_FILTER
7	TRIG_FILTER
8	TRIG_FILTER
9	TRIG_FILTER
10	TRIG_FILTER
11	TRIG_FILTER
12	TRIG_FILTER
13	TRIG_FILTER
14	TRIG_FILTER
15	TRIG_FILTER
16	TRIG_FILTER
17	TRIG_FILTER
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

TRIG_FILTER RO, TRIG_OPTS[17..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The trigger circuit includes a programmable noise filter. The value of this register controls the size of the noise pulse that will be considered noise and will be filtered out. Any pulses over this size will be consider signal. The units of this register are 4 nanoseconds. If this register is programmed to 0, nothing will be filtered out. If this register is programmed to 0x3ffff, pulses of up to 1 millisecond will be removed.

6.8 ENCA_OPTS

Bit	Name
0	ENCA_FILTER
1	ENCA_FILTER
2	ENCA_FILTER
3	ENCA_FILTER
4	ENCA_FILTER
5	ENCA_FILTER
6	ENCA_FILTER
7	ENCA_FILTER
8	ENCA_FILTER
9	ENCA_FILTER
10	ENCA_FILTER
11	ENCA_FILTER
12	ENCA_FILTER
13	ENCA_FILTER
14	ENCA_FILTER
15	ENCA_FILTER
16	ENCA_FILTER
17	ENCA_FILTER
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

ENCA_FILTER RO, ENCA_OPTS[17..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The encoder A circuit includes a programmable noise filter. The value of this register controls the size of the noise pulse that will be considered noise and will be filtered out. Any pulses over this size will be consider signal. The units of this register are 4 nanoseconds. If this register is programmed to 0, nothing will be filtered out. If this register is programmed to 0x3ffff, pulses of up to 1 millisecond will be removed.

6.9 ENCB_OPTS

Bit	Name
0	ENCB_FILTER
1	ENCB_FILTER
2	ENCB_FILTER
3	ENCB_FILTER
4	ENCB_FILTER
5	ENCB_FILTER
6	ENCB_FILTER
7	ENCB_FILTER
8	ENCB_FILTER
9	ENCB_FILTER
10	ENCB_FILTER
11	ENCB_FILTER
12	ENCB_FILTER
13	ENCB_FILTER
14	ENCB_FILTER
15	ENCB_FILTER
16	ENCB_FILTER
17	ENCB_FILTER
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

ENCB_FILTER RO, ENCB_OPTS[17..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The encoder B circuit includes a programmable noise filter. The value of this register controls the size of the noise pulse that will be considered noise and will be filtered out. Any pulses over this size will be consider signal. The units of this register are 4 nanoseconds. If this register is programmed to 0, nothing will be filtered out. If this register is programmed to 0x3ffff, pulses of up to 1 millisecond will be removed.

6.10 BOX_OUT_DYN_SEL_SET_A

Bit	Name
0	BOX_OUT_DYN_SEL_0
1	BOX_OUT_DYN_SEL_0
2	BOX_OUT_DYN_SEL_0
3	BOX_OUT_DYN_SEL_0
4	BOX_OUT_DYN_SEL_0
5	BOX_OUT_DYN_SEL_0
6	BOX_OUT_DYN_SEL_0
7	BOX_OUT_DYN_SEL_0
8	BOX_OUT_DYN_SEL_1
9	BOX_OUT_DYN_SEL_1
10	BOX_OUT_DYN_SEL_1
11	BOX_OUT_DYN_SEL_1
12	BOX_OUT_DYN_SEL_1
13	BOX_OUT_DYN_SEL_1
14	BOX_OUT_DYN_SEL_1
15	BOX_OUT_DYN_SEL_1
16	BOX_OUT_DYN_SEL_2
17	BOX_OUT_DYN_SEL_2
18	BOX_OUT_DYN_SEL_2
19	BOX_OUT_DYN_SEL_2
20	BOX_OUT_DYN_SEL_2
21	BOX_OUT_DYN_SEL_2
22	BOX_OUT_DYN_SEL_2
23	BOX_OUT_DYN_SEL_2
24	BOX_OUT_DYN_SEL_3
25	BOX_OUT_DYN_SEL_3
26	BOX_OUT_DYN_SEL_3
27	BOX_OUT_DYN_SEL_3
28	BOX_OUT_DYN_SEL_3
29	BOX_OUT_DYN_SEL_3
30	BOX_OUT_DYN_SEL_3
31	BOX_OUT_DYN_SEL_3

BOX_OUT_DYN_SEL_0 R/W, BOX_OUT_DYN_SEL_SET_A[7..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register controls the dynamic source for the three BitBox output signals BOX_OUT_DIFF_0, BOX_OUT_TTL_0 and BOX_OUT_OPTO_0. These three outputs always have the same dynamic source (they can individually be set to static high or low value). The following table lists all the possible sources.

Table 6-1 BitBox Dynamic Output Source Selections

BOX_OUT_DYN_SEL_0	Source
0 (00000b)	VFG0_CC1
1 (00001b)	VFG0_CC2
2 (00010b)	VFG0_CC3
3 (00011b)	VFG0_CC4
4 (00100b)	VFG1_CC1
5 (00101b)	VFG1_CC2
6 (00110b)	VFG1_CC3
7 (00111b)	VFG1_CC4
8 (01000b)	VFG2_CC1
9 (01001b)	VFG2_CC2
10 (01010b)	VFG2_CC3
11 (01011b)	VFG2_CC4
12 (01100b)	VFG3_CC1
13 (01101b)	VFG3_CC2
14 (01110b)	VFG3_CC3
15 (01111b)	VFG3_CC4
16 (10000b)	VFG0_ATS_CT0
17 (10001b)	VFG0_ATS_CT1
18 (10010b)	VFG0_ATS_CT2
19 (10011b)	VFG0_ATS_CT3
20 to 256	Reserved

Note: Not all of the sources above are available on all models. For example, the Axion 1xE only has one VFG, so sources with the prefix "VFG1", "VFG2" and "VFG3" will not be available. Similarly, a two VFG model (e.g. Axion-2xB) will not have the sources that start with "VF2" or "VFG3".

BOX_OUT_DYN_SEL_1 R/W, BOX_OUT_DYN_SEL_SET_A[15..8], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register controls the dynamic source for the three BitBox output signals BOX_OUT_DIFF_1, BOX_OUT_TTL_1 and BOX_OUT_OPTO_1. These three outputs always have the same dynamic source (they can individually be set to static high or low value). Table 6-1 lists all the possible sources.

BOX_OUT_DYN_SEL_2 R/W, BOX_OUT_DYN_SEL_SET_A[23..16], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register controls the dynamic source for the three BitBox output signals BOX_OUT_DIFF_2, BOX_OUT_TTL_2 and BOX_OUT_OPTO_2. These three outputs always have the same dynamic source (they can individually be set to static high or low value). Table 6-1 lists all the possible sources.

BOX_OUT_DYN_SEL_3 R/W, BOX_OUT_DYN_SEL_SET_A[31..24], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register controls the dynamic source for the three BitBox output signals BOX_OUT_DIFF_3, BOX_OUT_TTL_3 and BOX_OUT_OPTO_3. These three outputs always have the same dynamic source (they can individually be set to static high or low value). Table 6-1 lists all the possible sources.

6.11 BOX_OUT_DYN_SEL_SET_B

Bit	Name
0	BOX_OUT_DYN_SEL_4
1	BOX_OUT_DYN_SEL_4
2	BOX_OUT_DYN_SEL_4
3	BOX_OUT_DYN_SEL_4
4	BOX_OUT_DYN_SEL_4
5	BOX_OUT_DYN_SEL_4
6	BOX_OUT_DYN_SEL_4
7	BOX_OUT_DYN_SEL_4
8	BOX_OUT_DYN_SEL_5
9	BOX_OUT_DYN_SEL_5
10	BOX_OUT_DYN_SEL_5
11	BOX_OUT_DYN_SEL_5
12	BOX_OUT_DYN_SEL_5
13	BOX_OUT_DYN_SEL_5
14	BOX_OUT_DYN_SEL_5
15	BOX_OUT_DYN_SEL_5
16	BOX_OUT_DYN_SEL_6
17	BOX_OUT_DYN_SEL_6
18	BOX_OUT_DYN_SEL_6
19	BOX_OUT_DYN_SEL_6
20	BOX_OUT_DYN_SEL_6
21	BOX_OUT_DYN_SEL_6
22	BOX_OUT_DYN_SEL_6
23	BOX_OUT_DYN_SEL_6
24	BOX_OUT_DYN_SEL_7
25	BOX_OUT_DYN_SEL_7
26	BOX_OUT_DYN_SEL_7
27	BOX_OUT_DYN_SEL_7
28	BOX_OUT_DYN_SEL_7
29	BOX_OUT_DYN_SEL_7
30	BOX_OUT_DYN_SEL_7
31	BOX_OUT_DYN_SEL_7

BOX_OUT_DYN_SEL_4 R/W, BOX_OUT_DYN_SEL_SET_B[7..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register controls the dynamic source for the three BitBox output signals BOX_OUT_DIFF_4, BOX_OUT_TTL_4 and BOX_OUT_OPTO_4. These three outputs always have the same dynamic source (they can individually be set to static high or low value). Table 6-1 lists all the possible sources.

BOX_OUT_DYN_SEL_5 R/W, BOX_OUT_DYN_SEL_SET_B[15..8], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register controls the dynamic source for the three BitBox output signals BOX_OUT_DIFF_5, BOX_OUT_TTL_5 and BOX_OUT_OPTO_5. These three outputs always have the same dynamic source (they can individually be set to static high or low value). Table 6-1 lists all the possible sources.

BOX_OUT_DYN_SEL_6 R/W, BOX_OUT_DYN_SEL_SET_B[23..16], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register controls the dynamic source for the three BitBox output signals BOX_OUT_DIFF_6, BOX_OUT_TTL_6 and BOX_OUT_OPTO_6. These three outputs always have the same dynamic source (they can individually be set to static high or low value). Table 6-1 lists all the possible sources.

BOX_OUT_DYN_SEL_7 R/W, BOX_OUT_DYN_SEL_SET_B[31..24], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register controls the dynamic source for the three BitBox output signals BOX_OUT_DIFF_7, BOX_OUT_TTL_7 and BOX_OUT_OPTO_7. These three outputs always have the same dynamic source (they can individually be set to static high or low value). Table 6-1 lists all the possible sources.

6.12 BOX_OUT_DYN_SEL_SET_C

Bit	Name
0	BOX_OUT_DYN_SEL_8
1	BOX_OUT_DYN_SEL_8
2	BOX_OUT_DYN_SEL_8
3	BOX_OUT_DYN_SEL_8
4	BOX_OUT_DYN_SEL_8
5	BOX_OUT_DYN_SEL_8
6	BOX_OUT_DYN_SEL_8
7	BOX_OUT_DYN_SEL_8
8	BOX_OUT_DYN_SEL_9
9	BOX_OUT_DYN_SEL_9
10	BOX_OUT_DYN_SEL_9
11	BOX_OUT_DYN_SEL_9
12	BOX_OUT_DYN_SEL_9
13	BOX_OUT_DYN_SEL_9
14	BOX_OUT_DYN_SEL_9
15	BOX_OUT_DYN_SEL_9
16	BOX_OUT_DYN_SEL_10
17	BOX_OUT_DYN_SEL_10
18	BOX_OUT_DYN_SEL_10
19	BOX_OUT_DYN_SEL_10
20	BOX_OUT_DYN_SEL_10
21	BOX_OUT_DYN_SEL_10
22	BOX_OUT_DYN_SEL_10
23	BOX_OUT_DYN_SEL_10
24	BOX_OUT_DYN_SEL_11
25	BOX_OUT_DYN_SEL_11
26	BOX_OUT_DYN_SEL_11
27	BOX_OUT_DYN_SEL_11
28	BOX_OUT_DYN_SEL_11
29	BOX_OUT_DYN_SEL_11
30	BOX_OUT_DYN_SEL_11
31	BOX_OUT_DYN_SEL_11

BOX_OUT_DYN_SEL_8 R/W, BOX_OUT_DYN_SEL_SET_C[7..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register controls the dynamic source for the three BitBox output signals BOX_OUT_DIFF_8, BOX_OUT_TTL_8 and BOX_OUT_OC_0. These three outputs always have the same dynamic source (they can individually be set to static high or low value). Table 6-1 lists all the possible sources.

BOX_OUT_DYN_SEL_9 R/W, BOX_OUT_DYN_SEL_SET_C[15..8], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register controls the dynamic source for the three BitBox output signals BOX_OUT_DIFF_9, BOX_OUT_TTL_9 and BOX_OUT_OC_1. These three outputs always have the same dynamic source (they can individually be set to static high or low value). Table 6-1 lists all the possible sources.

BOX_OUT_DYN_SEL_10 R/W, BOX_OUT_DYN_SEL_SET_C[23..16], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register controls the dynamic source for the three BitBox output signals BOX_OUT_DIFF_10, BOX_OUT_TTL_10 and BOX_OC_2. These three outputs always have the same dynamic source (they can individually be set to static high or low value). Table 6-1 lists all the possible sources.

BOX_OUT_DYN_SEL_11 R/W, BOX_OUT_DYN_SEL_SET_C[31..24], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register controls the dynamic source for the three BitBox output signals BOX_OUT_DIFF_11, BOX_OUT_TTL_11 and BOX_OUT_OC_3. These three outputs always have the same dynamic source (they can individually be set to static high or low value). Table 6-1 lists all the possible sources.

6.13 BOX_OUT_MODE_SET_A

Bit	Name
0	BOX_OUT_MODE_TTL_0
1	BOX_OUT_MODE_TTL_0
2	BOX_OUT_MODE_TTL_1
3	BOX_OUT_MODE_TTL_1
4	BOX_OUT_MODE_TTL_2
5	BOX_OUT_MODE_TTL_2
6	BOX_OUT_MODE_TTL_3
7	BOX_OUT_MODE_TTL_3
8	BOX_OUT_MODE_TTL_4
9	BOX_OUT_MODE_TTL_4
10	BOX_OUT_MODE_TTL_5
11	BOX_OUT_MODE_TTL_5
12	BOX_OUT_MODE_TTL_6
13	BOX_OUT_MODE_TTL_6
14	BOX_OUT_MODE_TTL_7
15	BOX_OUT_MODE_TTL_7
16	BOX_OUT_MODE_TTL_8
17	BOX_OUT_MODE_TTL_8
18	BOX_OUT_MODE_TTL_9
19	BOX_OUT_MODE_TTL_9
20	BOX_OUT_MODE_TTL_10
21	BOX_OUT_MODE_TTL_10
22	BOX_OUT_MODE_TTL_11
23	BOX_OUT_MODE_TTL_11
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

BOX_OUT_MODE_TTL_0

R/W, BOX_OUT_MODE_SET_A[1..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_TTL_0. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_0 bitfield.

BOX_OUT_MODE_TTL_0	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_TTL_1

R/W, BOX_OUT_MODE_SET_A[3..2], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_TTL_1. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_1 bitfield.

BOX_OUT_MODE_TTL_1	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_TTL_2

R/W, BOX_OUT_MODE_SET_A[5..4], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_TTL_2. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_2 bitfield.

BOX_OUT_MODE_TTL_2	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_TTL_3 R/W, BOX_OUT_MODE_SET_A[7..6], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_TTL_3. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_3 bitfield.

BOX_OUT_MODE_TTL_3	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_TTL_4 R/W, BOX_OUT_MODE_SET_A[9..8], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_TTL_4. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_4 bitfield.

BOX_OUT_MODE_TTL_$	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_TTL_5 R/W, BOX_OUT_MODE_SET_A[11..10], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_TTL_5. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_5 bitfield.

BOX_OUT_MODE_TTL_5	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_TTL_6 R/W, BOX_OUT_MODE_SET_A[13..12], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_TTL_6. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_6 bitfield.

BOX_OUT_MODE_TTL_6	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_TTL_7 R/W, BOX_OUT_MODE_SET_A[15..14], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_TTL_7. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_7 bitfield.

BOX_OUT_MODE_TTL_7	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_TTL_8 R/W, BOX_OUT_MODE_SET_A[17..16], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_TTL_8. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_8 bitfield.

BOX_OUT_MODE_TTL_8	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_TTL_9 R/W, BOX_OUT_MODE_SET_A[19..18], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_TTL_9. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_9 bitfield.

BOX_OUT_MODE_TTL_9	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_TTL_10 R/W, BOX_OUT_MODE_SET_A[21..20], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_TTL_10. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_10 bitfield.

BOX_OUT_MODE_TTL_10	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_TTL_11 R/W, BOX_OUT_MODE_SET_A[23..22], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_TTL_11. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_11 bitfield.

BOX_OUT_MODE_TTL_11	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

6.14 BOX_OUT_MODE_SET_B

Bit	Name
0	BOX_OUT_MODE_DIFF_0
1	BOX_OUT_MODE_DIFF_0
2	BOX_OUT_MODE_DIFF_1
3	BOX_OUT_MODE_DIFF_1
4	BOX_OUT_MODE_DIFF_2
5	BOX_OUT_MODE_DIFF_2
6	BOX_OUT_MODE_DIFF_3
7	BOX_OUT_MODE_DIFF_3
8	BOX_OUT_MODE_DIFF_4
9	BOX_OUT_MODE_DIFF_4
10	BOX_OUT_MODE_DIFF_5
11	BOX_OUT_MODE_DIFF_5
12	BOX_OUT_MODE_DIFF_6
13	BOX_OUT_MODE_DIFF_6
14	BOX_OUT_MODE_DIFF_7
15	BOX_OUT_MODE_DIFF_7
16	BOX_OUT_MODE_DIFF_8
17	BOX_OUT_MODE_DIFF_8
18	BOX_OUT_MODE_DIFF_9
19	BOX_OUT_MODE_DIFF_9
20	BOX_OUT_MODE_DIFF_10
21	BOX_OUT_MODE_DIFF_10
22	BOX_OUT_MODE_DIFF_11
23	BOX_OUT_MODE_DIFF_11
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

BOX_OUT_MODE_DIFF_0 R/W, BOX_OUT_MODE_SET_B[1..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_DIFF_0. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_0 bitfield.

BOX_OUT_MODE_DIFF_0	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_DIFF_1 R/W, BOX_OUT_MODE_SET_B[3..2], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_DIFF_1. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_1 bitfield.

BOX_OUT_MODE_DIFF_1	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_DIFF_2 R/W, BOX_OUT_MODE_SET_B[5..4], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_DIFF_2. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_2 bitfield.

BOX_OUT_MODE_DIFF_2	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_DIFF_3 R/W, BOX_OUT_MODE_SET_B[7..6], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_DIFF_3 . This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_d bitfield.

BOX_OUT_MODE_DIFF_3	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_DIFF_4 R/W, BOX_OUT_MODE_SET_B[9..8], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_DIFF_4. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_4 bitfield.

BOX_OUT_MODE_DIFF_4	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_DIFF_5 R/W, BOX_OUT_MODE_SET_B[11..10], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_DIFF_5. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_5 bitfield.

BOX_OUT_MODE_DIFF_5	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_DIFF_6 R/W, BOX_OUT_MODE_SET_B[13..12], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_DIFF_6. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_6 bitfield.

BOX_OUT_MODE_DIFF_6	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_DIFF_7 R/W, BOX_OUT_MODE_SET_B[15..14], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_DIFF_7. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_7 bitfield.

BOX_OUT_MODE_DIFF_7	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_DIFF_8 R/W, BOX_OUT_MODE_SET_B[17..16], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_DIFF_8. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_8 bitfield.

BOX_OUT_MODE_DIFF_8	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_DIFF_9 R/W, BOX_OUT_MODE_SET_B[19..18], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_DIFF_9. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_9bitfield.

BOX_OUT_MODE_DIFF_9	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_DIFF_10 R/W, BOX_OUT_MODE_SET_B[21..20], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_DIFF_10. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_10 bitfield.

BOX_OUT_MODE_DIFF_10	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_DIFF_11 R/W, BOX_OUT_MODE_SET_B[23..22], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_DIFF_11. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_11 bitfield.

BOX_OUT_MODE_DIFF_11	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

6.15 BOX_OUT_MODE_SET_C

Bit	Name
0	BOX_OUT_MODE_OPTO_0
1	BOX_OUT_MODE_OPTO_0
2	BOX_OUT_MODE_OPTO_1
3	BOX_OUT_MODE_OPTO_1
4	BOX_OUT_MODE_OPTO_2
5	BOX_OUT_MODE_OPTO_2
6	BOX_OUT_MODE_OPTO_3
7	BOX_OUT_MODE_OPTO_3
8	BOX_OUT_MODE_OPTO_4
9	BOX_OUT_MODE_OPTO_4
10	BOX_OUT_MODE_OPTO_5
11	BOX_OUT_MODE_OPTO_5
12	BOX_OUT_MODE_OPTO_6
13	BOX_OUT_MODE_OPTO_6
14	BOX_OUT_MODE_OPTO_7
15	BOX_OUT_MODE_OPTO_7
16	BOX_OUT_MODE_OC_0
17	BOX_OUT_MODE_OC_0
18	BOX_OUT_MODE_OC_1
19	BOX_OUT_MODE_OC_1
20	BOX_OUT_MODE_OC_2
21	BOX_OUT_MODE_OC_2
22	BOX_OUT_MODE_OC_3
23	BOX_OUT_MODE_OC_3
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

BOX_OUT_MODE_OPTO_0 R/W, BOX_OUT_MODE_SET_C[1..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_OPTO_0. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_0 bitfield.

BOX_OUT_MODE_OPTO_0	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_OPTO_1 R/W, BOX_OUT_MODE_SET_C[3..2], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_OPTO_1 This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_1 bitfield.

BOX_OUT_MODE_OPTO_1	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_OPTO_2 R/W, BOX_OUT_MODE_SET_C[5..4], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_OPTO_2 This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_2 bitfield.

BOX_OUT_MODE_OPTO_2	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_OPTO_3 R/W, BOX_OUT_MODE_SET_C[7..6], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_OPTO_3. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_3 bitfield.

BOX_OUT_MODE_XXX_3	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_OPTO_4 R/W, BOX_OUT_MODE_SET_C[9..8], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_OPTO_4. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_4 bitfield.

BOX_OUT_MODE_OPTO_4	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_OPTO_5 R/W, BOX_OUT_MODE_SET_C[11..10], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_OPTO_5. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_5 bitfield.

BOX_OUT_MODE_OPTO_5	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_OPTO_6 R/W, BOX_OUT_MODE_SET_C[13..12], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_OPTO_6. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_6 bitfield.

BOX_OUT_MODE_OPTO_6	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_OPTO_7 R/W, BOX_OUT_MODE_SET_C[15..14], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_OPTO_7. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_7 bitfield.

BOX_OUT_MODE_OPTO_7	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_OC_0 R/W, BOX_OUT_MODE_SET_C[17..16], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_OC_0 This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_8 bitfield.

BOX_OUT_MODE_OC_0	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_OC_1 R/W, BOX_OUT_MODE_SET_C[19..18], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_OC_1. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_9 bitfield.

BOX_OUT_MODE_OC_1	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_OC_2 R/W, BOX_OUT_MODE_SET_C[21..20], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_OC_2. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_10 bitfield.

BOX_OUT_MODE_OC_2	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

BOX_OUT_MODE_OC_3 R/W, BOX_OUT_MODE_SET_C[23..22], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the BitBox output BOX_OUT_MODE_OC_3. This bit controls whether this output is static (high or low) or dynamic. If the bit is dynamic, the source is controlled by the associated BOX_OUT_DYN_SEL_11 bitfield.

BOX_OUT_MODE_OC_3	Meaning
0 (00b)	Static output low
1 (01b)	Static output high
2 (10b)	Dynamic output
3 (11b)	Reserved

Encoder Divider

Chapter 7

7.1 Introduction

This section covers the encoder divider which supported on all of BitFlow's modern frame grabber families. The purpose of Encoder Divider is to provide the ability to use an encoder running at one rate to drive a line scan camera at a different rate. This circuit is only useful for line scan cameras. The Encoder Divider can scale up or down the incoming encoder frequency. The encoder divider is fully programmable and is easily controlled from software and/or from camera configuration files.

The factor used to scaled the incoming encoder frequency does not have to be a whole number. For example, the encoder could be scaled by 0.03448 or 4.2666). Of course, not all rational numbers in the available scaling range can be selected (there are an infinite number of them). However, a useful selection of values is available which should support most applications.

The Encoder Divider circuit takes as its input the selected encoder input. The output of the encoder divider drives the same parts of the board the normal encoder usually does. The actual circuit(s) being driven depends on how the board is programmed.

7.2 Encoder Divider Details

7.2.1 Formula

The following formula shows the equation used to scale the encoming encoder rate into the camera's line rate:

$$F_{out} = F_{in} \frac{2^N}{M}$$

W:

Fout = The frequency used to driver the camera or the NTG or the CTabs
Fin = The encoder (input) frequency
N = An integer between 0 and 6 (set by the register ENC_DIV_N)
M = An integer between 1 and 1023 (set by the register ENC_DIV_M)

The above formula provides an effective scaling factor from 0.001 (N = 0, M =1023) to 64 (N = 6, M = 1). Not every scaling factor can be achieved between these two extremes, and the scaling factors are not evenly distributed. However, a scaling factor can be generally found that meets the requirements of most applications.

7.2.2 Example

Let's assume that the encoder frequency (Fin) is 10 KHz and that we need an output (Fout) of ~30 KHz. This means that we need to multiply by 3. Set N = 6 and M = 21. This will a scaling factor of 3.048. The result is an effective line rate of 30.48 KHz.

7.2.3 Restrictions

Because the encoder divider uses a digital PLL run by a high frequency clock, not all encoder input frequencies can be accurately scaled. The PLL has been designed to work in most machine visions applications. Support, therefore, is provided for the following input frequency range:

Minimum input encoder frequency: 1.6 KHz
Maximum input encoder frequency: 300 KHz

7.2.4 PLL Locking

The encoder divider achieves its scaling using a PLL. By default the output waveform is locked to the input waveform. However, this locking can result in a small amount of jitter. To reduce the jitter, the output waveform can be run open loop. This mode is accessed by setting the register ENC_DIV_OPEN_LOOP to 1.

7.2.5 Handling Encoder Slow Down or Stopping

On some machine vision systems, the encoder is attached to a mechanism that may slow and/or stop. Any PLL has a limited range that it can track (based on the PLL master clock), outside of this range, the output signal can become unpredictable. The Encoder Divider circuit's master clock is 50 MHz, which makes the minimum frequency that it can accurately track around 1.6 KHz. In order to avoid this situation and handle encoder slow down/stop gracefully, the encoder divider has has limiting circuit that can be run in one of two different mode described in the following two sections.

Slow Tracking Mode (ENC_DIC_FORCE_DC = 0)

In this mode, when the input frequency goes below the minimum of 1.6 KHz, the Encoder Divider circuit's output will continue to track the input, but the output frequency will become simple divider on the input freqency. In this mode the output will track the input using the following formula (the variables are the same as in Section 7.2.1)

$$F_{out} = \frac{F_{in}}{4M}$$

DC Mode (ENC_DIV_FORCE_DC = 1)

In this mode, the Fin goes below 1.6 KHz, Fout will goes to DC. This means that when the input frequency goes below the minimum the camera will be frozen, acquisition will stop. The board will stay in this state until Fin goes above 1.6 KHz. This is useful when the encoder is being driven by a stage that is traveling back and forth. At both ends of travel when the stage changes directions, the board will not acquire.

7.3 Encoder Divider Control Registers

The following table summarizes the registers:

Table 7-1 Encoder Divider Registers

Name	Purpose
ENC_DIV_M	This controls the M factor in the Encoder Divider equation (see Section 7.2.1)
ENC_DIV_N	The controls the N factor the Encoder Divider equation
ENC_DIV_FORCE_DC	Controls the behavior when Fin falls below the minimum. 0 = Output runs in simple divider mode. 1 = Output goes to DC.
ENC_DIV_OPEN_LOOP	Controls whether the output signal phase of the Encoder Divider is lock to the intput or is allowed to free run. 0 = Output phased locked to input. 1 = Ouput runs open loop.
ENC_DIV_FCLK_SEL	Reserved for future support for alternate Encoder Divider PLL Master clock frequencies. Currently must be set to 0, which selects 50 MHz clock.

See Chapter 9 for details on the registers needed to control the encoder divider system.

Quadrature Encoder

Chapter 8

8.1 Introduction

This section discusses support for quadrature encoders. A quadrature encoder is an encoder that outputs two signals A and B. Both signals are used as a line trigger. However, the signals are 90 degrees out of phase. By comparing the A and B signals, the direction of the encoder motion can be determined. There are a number of ways that quadrature encoders can be used to control acquisition. The following sections cover all of the support methods.

Most of the quadrature encoder system is based around a 24-bit counter. This normally starts at zero and then counts up or down every time the encoder moves. The counter can be observed at any time via the QENC_COUNT register. This registers is the heart of the encoder system. For example, triggers values can be programmed to start and end acquisition of lines. Also, as the counter tracks the motion of the stage attached to the encoder exactly, the system can be programmed to acquire forward only or backward only stage movements. The system can be programmed to only acquire one line for each encoder count that corresponds to a physical location on the stage. The encoder counter can be used in many different ways, described in more details below.

8.1.1 Simple Encoder Mode

The most basic method of using a quadrature encoder is to use it like a standard signal phase encoder. In this mode, the quadrature encoder provides a higher resolution signal, as both the A and B signals can be used to trigger lines. Also, by setting QENC_DECODE = 1, both the rising and the falling edges of both the A and B signals are used to trigger lines, providing a 4x increase in resolution over a signal phase encoder.

In this mode, every encoder edge triggers a line, the direction information from the encoder is ignored.

8.1.2 Positive or Negative Only Acquisition

The board can be programmed to only acquired lines when the encoder moves forward (increase the encoder count in a positive direction) or moves backwards (decrease the encoder count in a negative direction). This mode is useful in situations where a stage is moving back and forth, and lines need only be acquired while the stage is moving in one direction, and not the other direction. The direction of acquisition is controlled by the QENC_AQ_DIR register.

8.1.3 Interval Mode

Often in situations when a stage is moving back and forth, acquisition is only required over a subsection of the total stage range. Interval mode has been designed for these situations. When the board is in interval mode, it only acquires lines when the encoder counter is between a lower limit and an upper limit. If the counter is outside these limits, lines are not acquired.

To use interval mode, set QENC_INTRVL_MODE = 1, and program QENC_INTRVL_LL and QENC_INTRVL_UL to the encoder ranges that bracket the section of your stage range that you wish to acquire. Interval mode can be used in conjunction with QENC_AQ_DIR to acquire lines passing over the interval in the positive direction, the negative direction or both directions.

8.1.4 Re-Acquisition Prevention

Encoders are usually connected to mechanical systems which do not always move smoothly. Because of these imperfections, t can be “jitter” in the quadrature encoder signal. This jitter is not an electrical imperfection, but represent the reality of the mechanical system vibrating, jumping, bouncing, etc. If these imperfections occur during the period of time w lines are being acquired, the image will be distorted. Lines on the object can be acquired more than once as the stage jitters. To prevent re-acquisition of lines, a circuit has been added to the quadrature encoder system that can prevent any line from being acquired more than once. To enable this mode, set QENC_NO_REAQ = 1.

8.1.5 Scan Step Mode

The encoder can also be used to trigger acquisition of full frames from an area scan camera. The idea is that every N lines, a trigger is issued to the board, which causes acquisition of a frame. This can be used, for example, with a linear stage, w an image is needed in steps across the range of the stage. This mode is enable by setting SCAN_STEP_TRIG = 1, and programming SCAN_STEP to the number of encoder counts per trigger.

8.1.6 Combining Modes

All of the modes above can be combined to support complicated encoder requirements. For example, the board can be programmed to acquire an interval in the positive direction only, with no lines being reacquired. Many combinations are possible.

8.1.7 Control Registers

See Chapter 9 for the registers needed to control the quadrature encoder system.

8.1.8 Observability

The status of the quadrature encoder system can be observed at any time. Shown in Table 8-1are all the registers that can be used.:

Table 8-1 Observability Registers.

Register	Meaning
QENC_COUNT	Encoder counter
QENC_PHASEA	Phase of input A
QENC_PHASEB	Phase of input B
QENC_DIR	Direction of encoder
QENC_INTRVL_IN	Interval status
QENC_NEW_LINES	Indicates new lines are being acquired

8.1.9 Electrical Connections

Both TTL and LVDS (differential) quadrature encoders are supported. TTL connections are shown in Table 8-2 and LVDS connections are shown in Table 8-3.

Table 8-2 TTL Quadrature Encoder Connections

Encoder	Frame Grabber
A	VFGx_ENCODERA_TTL
B	VFGx_ENCODERB_TTL
Ground	GND

Table 8-3 LVDS Quadrature Encoder Connections

Encoder	Frame Grabber
A+	VFGx_ENCODERA+
A-	VFGx_ENCODERA-
B+	VFGx_ENCODERB+
B-	VFGx_ENCODERB-

Note: VFGx - refers to the VFG number that you wish to connect to. For example, if you want to connect a TLL A output to VFG 0, then you would use VFG0_ENCODERA_TTL.

8.2 Understanding Stage Movement vs. Quadrature Encoder Modes

The quadrature encoder system has many modes that can be used in various combinations. These combinations are easier to understand through a few simple illustrations. Figure 8-1 shows the basic Encoder Count vs. Time graph and how it corresponds to stage movement. Keep in mind that the encoder could be attached to any mechanical system, however, a back and forth stage is a simple way to illustrate these modes.

In Figure 8-1 you can see as the stage moves back and forth, the encoder counts up and down. Further, in this example we assume QENC_AQ_DIR = 1, which tells the system to only acquire when the encoder counter is moving in the positive direction. This is illustrated by solid lines in the positive direction and dashed lines in the negative direction.

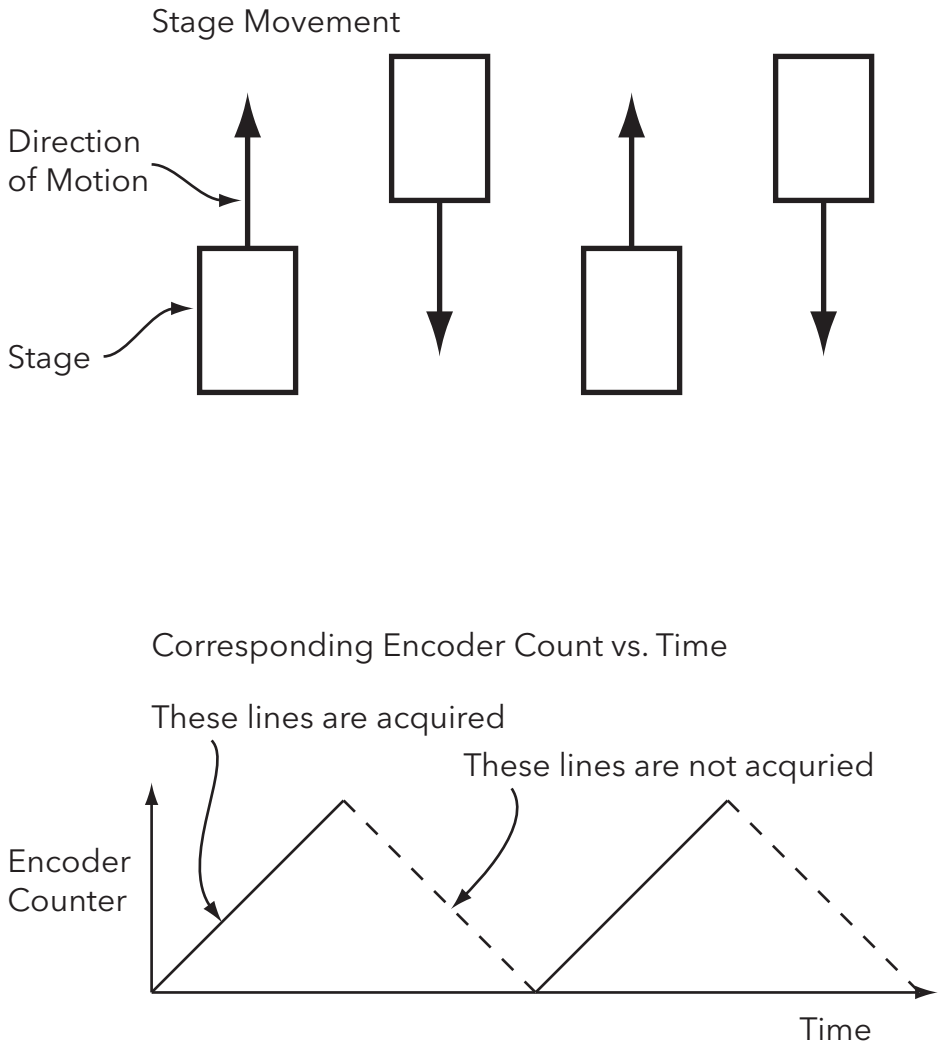

Figure 8-1 Encoder Count vs Time

Figure 8-2 shows all of the major quadrature encoder modes.

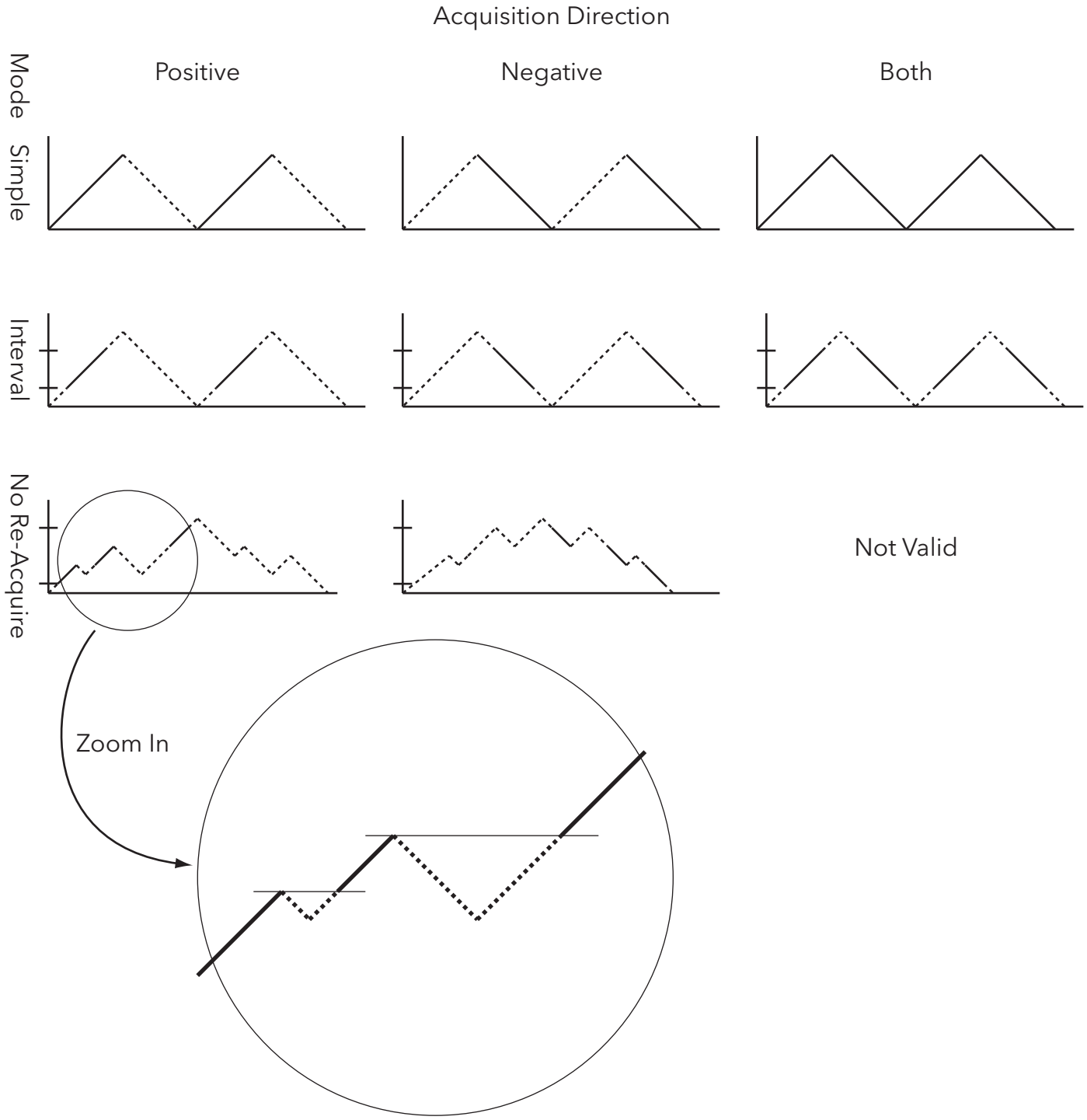

Figure 8-2 Quadrature Encoder Modes vs. Acquisition

Quadrature Encoder and Divider Registers

Chapter 9

9.1 Introduction

This section enumerates the registers used to control the boards quadrature encoder circuit and encoder divider circuit.

9.2 CON65 Register

Bit	Name
0	SEL_ENCQ
1	SEL_ENCDIV_INPUT
2	SEL_ENCDIV
3	ENC_DIV_N
4	ENC_DIV_N
5	ENC_DIV_N
6	ENC_DIV_M
7	ENC_DIV_M
8	ENC_DIV_M
9	ENC_DIV_M
10	ENC_DIV_M
11	ENC_DIV_M
12	ENC_DIV_M
13	ENC_DIV_M
14	ENC_DIV_M
15	ENC_DIV_M
16	SCAN_STEP
17	SCAN_STEP
18	SCAN_STEP
19	SCAN_STEP
20	SCAN_STEP
21	SCAN_STEP
22	SCAN_STEP
23	SCAN_STEP
24	SCAN_STEP
25	SCAN_STEP
26	SCAN_STEP
27	SCAN_STEP
28	SCAN_STEP
29	SCAN_STEP
30	SCAN_STEP
31	SCAN_STEP

SEL_ENCQ R/W, CON65[0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit selects which quadrature encoder circuit output will be used on this VFG.

SEL_ENCQ	Meaning
0	Select the output of this VFG's quadrature circuit output
1	Select the output of VFG0's quadrature circuit output

SEL_ENCDIV_INPUT R/W, CON65[1], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit selects which input will drive the encoder divider circuit.

SEL_ENC_DIV_INPUT	Meaning
0	VFGx_ENCA_SEL
1	The output of this VFG's quadrature circuit output

SEL_ENCDIV R/W, CON65[2], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit selects which encoder divider circuit output will be used on this VFG.

SEL_ENCDIV	Meaning
0	Select the output of this VFG's encoder divider
1	Select the output of VFG0's encoder divider

ENC_DIV_N R/W, CON65[5..3], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

These bits set the N part of the encoder divider equation. See Section 7.1 for more information.

ENC_DIV_M R/W, CON65[15..6], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

These bits set the M part of the encoder divider equation. See Section 7.1 for more information.

SCAN_STEP R/W, CON65[31..16], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bitfield controls the number of encoder pulses that must occur before a trigger is issued to the system. See SCAN_STEP_TRIG for more information. The Scan Step circuit takes into account the interval and re-acquisition functions.

9.3 CON66 Register

Bit	Name
0	QENC_INTRVL_LL
1	QENC_INTRVL_LL
2	QENC_INTRVL_LL
3	QENC_INTRVL_LL
4	QENC_INTRVL_LL
5	QENC_INTRVL_LL
6	QENC_INTRVL_LL
7	QENC_INTRVL_LL
8	QENC_INTRVL_LL
9	QENC_INTRVL_LL
10	QENC_INTRVL_LL
11	QENC_INTRVL_LL
12	QENC_INTRVL_LL
13	QENC_INTRVL_LL
14	QENC_INTRVL_LL
15	QENC_INTRVL_LL
16	QENC_INTRVL_LL
17	QENC_INTRVL_LL
18	QENC_INTRVL_LL
19	QENC_INTRVL_LL
20	QENC_INTRVL_LL
21	QENC_INTRVL_LL
22	QENC_INTRVL_LL
23	QENC_INTRVL_LL
24	QENC_DECODE
25	QENC_AQ_DIR
26	QENC_AQ_DIR
27	QENC_INTRVL_MODE
28	QENC_NO_REAQ
29	Reserved
30	SCAN_STEP_TRIG
31	QENC_RESET

QENC_INTRVL_LL R/WR/W, CON66[23..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register contains the lower limit value that is used to start acquisition when the system is in interval mode (see QENC_INTRVL_MODE).

QENC_DECODE R/W, CON66[24], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit determines how often the quadrature counter is incremented.

QENC_DECODE	Meaning
0	Counter increments on the rising edge of input A and the rising edge of input B. This is also called "2x" mode.
1	Counter increments on both the rising and falling edge of A and both the rising and falling edge of B. This is also called "4x" mode.

QENC_AQ_DIR R/W, CON66[26..25], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls which quadrature encoder direction is used for acquisition.

QENC_AQ_DIR	Meaning
0 (00b)	Lines are acquired in both directions
1 (01b)	Lines are acquired only in the positive direction.
2 (10b)	Lines are acquired only in the negative direction.
3 (11b)	Reserved

QENC_INTRVL_MODE R/W, CON66[27], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

When this bit is 1, interval mode is turned on. When interval mode is on, lines are only captured when the encoder counter is between the lower limit (set by QENC_INTRVL_LL) and the upper limit (set by QENT_INTRVL_UL). If the counter is outside of this range, lines are not acquired. Whether lines are acquired as the counter increments through the interval, or decrements through the interval, or in both directions are controlled by QENC_AQ_DIR.

QENC_NO_REAQ R/W, CON66[28], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls how the quadrature encoder system handles the situation w the encoder does not smoothly increase (or decrease if QENC_AQ_DIR = 1). If there is "jitter" in the encoder signal, often caused by problems with the mechanical systems,

it is possible for the board to acquire the same line or lines more than once as the mechanical system backs up and moves forward (jitter). This re-acquisition can cause problems as the resulting images will have distortions and will not accurately represent the object in front of the camera.

Programming this bit to a 1 turns on the no-reacquisition circuit. This circuit eliminates this problem as each line in the image will only be acquired once, regardless of how much jitter occurs in the quadrature encoder input. The circuit does this by making sure that only one line is acquired for each encoder counter value. If the quadrature encoder backs up, and then moves forward, the board will not acquire lines until a new encoder counter value is reached.

This system handles any amount of jitter, regardless of how many times the counter passes through a value, or to what extremes the counter goes. New lines will only be acquired when new values are reached.

Once the entire frame has been acquired, the system must be reset. The system can always be reset by poking QENC_RESET to 1. There are also ways that the system can automatically be reset, see QENC_RESET_MODE.

QENC_NO_REAQ	Meaning
0	Lines are acquired every change in the encoder counter (as controlled by QENC_AQ_DIR)
1	Lines are only acquired when the encoder counter reaches new values (also controlled by QENC_AQ_DIR)

SCAN_STEP_TRIG R/W, CON66[30], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

The scan step circuit uses the encoder to generate a trigger to the system. The scan step trigger generates a trigger every N lines (N is set in the SCAN_STEP register).

SCAN_STEP_TRIG	Meaning
0	The internal signal VFGx_ENCQ_SEL is the output of the quadrature encoder circuit
1	The internal signal VFGx_ENCQ_SEL is the output of the scan step circuit

QENC_RESET WO, CON66[31], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Poking this bit to a 1 resets the entire quadrature encoder system.

9.4 CON67 Register

Bit	Name
0	QENC_INTRVL_UL
1	QENC_INTRVL_UL
2	QENC_INTRVL_UL
3	QENC_INTRVL_UL
4	QENC_INTRVL_UL
5	QENC_INTRVL_UL
6	QENC_INTRVL_UL
7	QENC_INTRVL_UL
8	QENC_INTRVL_UL
9	QENC_INTRVL_UL
10	QENC_INTRVL_UL
11	QENC_INTRVL_UL
12	QENC_INTRVL_UL
13	QENC_INTRVL_UL
14	QENC_INTRVL_UL
15	QENC_INTRVL_UL
16	QENC_INTRVL_UL
17	QENC_INTRVL_UL
18	QENC_INTRVL_UL
19	QENC_INTRVL_UL
20	QENC_INTRVL_UL
21	QENC_INTRVL_UL
22	QENC_INTRVL_UL
23	QENC_INTRVL_UL
24	QENC_REAQ_MODE
25	QENC_REAQ_MODE
26	QENC_RESET_REAQ
27	ENC_DIV_FOURCE_DC
28	ENC_DIV_OPEN_LOOP
29	ENC_DIV_FCLK_SEL
30	ENC_DIV_FCLK_SEL
31	ENC_DIV_FCLK_SEL

QENC_INTRVL_UL R/W, CON67[23..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register contains the upper limit value that is used to start acquisition when the system is in interval mode (see QENC_INTRVL_MODE).

QENC_RESET_MODE R/W, CON67[25..24], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls how the circuit that prevents re-acquisition from encoder jitter is reset. Re-acquisition is prevented by keeping a list of lines that have been acquired, and making sure that only lines that are not on the list are acquired. Once the entire frame is acquired, t must be some way to reset the list, otherwise no new lines will ever be acquired See QENC_NO_REAQ for more information.

The reset can be either automatic or manual. Manual modes require that the host application software poke the QENC_RESET_REAQ bit when the reset is desired. Automatic modes do not require host interaction, the reset will occur automatically when the specified conditions are met.

QENC_REAQ_MODE	Mode	Meaning
0 (00b)	Manual	Reset the list of acquired lines when QENC_RESET_REAQ is poked to 1.
1 (01b)	Automatic	Reset the list of lines when the encoder counter is outside of the interval set by the upper limit and lower limit. Whether the reset occurs above the upper limit or below the lower limit depends on the QENC_AQ_DIR register.
2 (10b)		Reserved
3 (11b)		Reserved

QENC_RESET_REAQ WO, CON67[26], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This register is used to reset the circuit that prevents the re-acquisition of lines when QENC_NO_REAQ is set to 1. Writing a 1 to this register deletes the list of acquired lines, thus next time the lines are passed over, they will be acquired again. Writing to this bit always resets the no re-acquisition circuit, regardless of the mode set by the QENC_REAQ_MODE. However, the register QENC_REAQ_MODE can be used to set the board in a mode w the no re-acquisition circuit is reset automatically every pass over the image.

ENC_DIV_FORCE_DC R/W, CON67[27], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Setting this bit to 1 forces the encoder divider to output a DC signal when the input signal falls below a certain frequency.

ENC_DIV_OPEN_LOOP R/W, CON67[28], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Setting this bit to 1 forces the encoder divider to run open loop.

ENC_DIV_FCLK_SEL R/W, CON67[31..29], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Reserved for future support for alternate Encoder Divider PLL Master clock frequencies. Currently must be set to 0, which selects 125 MHz clock.

9.5 CON68 Register

Bit	Name
0	RD_ENCQ_SELECTED
1	RD_ENCDIV_SELECTED
2	ENC_DIV_RESET
3	ENC_DIV_AUTO_RESET_DISABLE
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

RD_ENCQ_SELECTED RO, CON68[0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit indicates the current state of selected quad encoder circuit output.

RD_ENCDIV_SELECTED RO, CON68[1], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit displays the current state of the selected encoder divider output.

ENC_DIV_RESET WO, CON68[2], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Writing a1 to this register resets the encoder divider/multiplier.

ENC_DIV_AUTO_RESET_DISABLE R/W, CON68[3], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Set this bit to 1 to disable the encoder divider/multiplier auto-reset function.

9.6 CON69 Register

Bit	Name
0	QENC_COUNT
1	QENC_COUNT
2	QENC_COUNT
3	QENC_COUNT
4	QENC_COUNT
5	QENC_COUNT
6	QENC_COUNT
7	QENC_COUNT
8	QENC_COUNT
9	QENC_COUNT
10	QENC_COUNT
11	QENC_COUNT
12	QENC_COUNT
13	QENC_COUNT
14	QENC_COUNT
15	QENC_COUNT
16	QENC_COUNT
17	QENC_COUNT
18	QENC_COUNT
19	QENC_COUNT
20	QENC_COUNT
21	QENC_COUNT
22	QENC_COUNT
23	QENC_COUNT
24	QENC_PHASEA
25	QENC_PHASEB
26	QENC_DIR
27	QENC_INTRVL_IN
28	QENC_NEW_LINES
29	Reserved
30	QENC_FREQ
31	QENC_FREQ

QENC_COUNT RO, CON69[23..0], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bitfield displays the current quadrature encoder count.

QENC_PHASEA RO, CON69[24], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit displays the current logic level of the A quadrature encoder phase.

QENC_PHASEB RO, CON69[25], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit displays the current logic level of the B quadrature encoder phase.

QENC_DIR RO, CON69[26], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit displays the current quadrature encoder direction.

QENC_DIR	Meaning
0	Direction is negative
1	Direction is positive

QENC_INTRVL_IN RO, CON69[27], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit indicates the current status of the quadrature encoder if the system is in interval mode (see QENC_INTRVL_MODE).

QENC_INTRVL_IN	Meaning
0	System is not inside the interval. Encoder counter is not between QENC_INTRVL_LL and QENC_INTRVL_UL. Lines are not being acquired.
1	System is inside the interval. Encoder counter is between QENC_INTRVL_LL and QENC_INTRVL_UL. Lines are being acquired.

QENC_NEW_LINES RO, CON69[28], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit indicates if the system is at an encoder count that corresponds to a new line. When QENC_NO_REAQ = 1, only lines that have not yet been scanned are acquired. This bit can be used to determine of new lines are being traversed, or if the system has backed up, and is revisiting old lines.

QENC_NEW_LINES	Meaning
0	The system is traversing lines that have already been visited. If QENC_NO_REAQ = 1, lines are not being acquired.
1	The system is traversing new lines. Lines are being acquired.

QENC_FREQ RO, CON69[31..30], Aon-CXP, Axion-CL, Claxon-CXP, Claxon-FXP, Cyton-CXP

Reserved for future support for alternate decoder timing.

CXP Subsystem Registers

Chapter 10

10.1 Introduction

The sections describe the registers used to access the CoaXPress engine on the Aon, Claxon and the Cyton. Most of these registers are programmed automatically by the software and are probably only need be accessed under very special circumstances.

There is one set of CXP registers for each link. For example, the Claxon-CXP4 has four sets of these registers, while the Aon-CXP1 has only one set. Each link has its own SERDES and associated decode circuitry. In addition, each link can be powered separately. For this reason there are four identical sets of registers, one for each link. These link registers are preceded by the link number in order to identify which link they belong to (see Table 10-1 for example).

Table 10-1 Link Register Example

Register Example Name	CXP Link number
0_POCXP_EN_POWER	0
1_POCXP_EN_POWER	1
2_POCXP_EN_POWER	2
3_POCXP_EN_POWER	3

It is also important to understand the each VFG has a complete set of these registers. This means that each VFG on the Cyton-CXP can access all of the links mounted on the board. The reason is that cameras can be single, dual or quad link. One VFG typically controls one camera, so each VFG needs access to all the links that belong to the connected camera.

To illustrate this concept, lets look at the Cyton-CXP4. This has 4 CXP links and 4 VFGs. There are quite a few ways that the cameras can be connected to VFGs. This board could be connected to four single link cameras, in which case each VFG gets data from one CXP link. It could also be connected to two dual link cameras. In this case, two VFGs get two CXP links each, and there are two VFGs which are not used and get no data. The final example is a four CXP link camera, in this case one VFG gets data from all four links, and the other three VFGs are not used and get no data. The board is quite flexible, but the software tries to keep things simple and make sure users to get into too much trouble. In general, the connects between CXP links and VFGs is automatic, and the user does not have to worry about which links go to which VFG.

Note: One boards with two CXP links, all of the link control registers will be available. However, only those registers that correspond to an physical link available on the board will do anything.

10.2 CON104

Bit	Name
0	0_POCXP_EN_POWER
1	0_POCXP_EN_24V_REG
2	0_POCXP_EN_CAM_SENSE
3	0_POCXP_CAM_IS_POCXP
4	0_POCXP_SHORT_DETECTED
5	0_POCXP_OPEN_DETECTED
6	0_POCXP_OVER_DETECTED
7	0_POCXP_OVER_LATCH
8	0_POCXP_UNDER_DETECTED
9	0_POCXP_UNDER_LATCH
10	0_POCXP_24V_OK
11	Reserved
12	0_POCXP_STATE
13	0_POCXP_OVR_AUTO_RESTART
14	0_POCXP_SENSE_BYPASS
15	0_ENABLE_POCXP_SYSTEM
16	0_POCXP_CURRENT_LATCH
17	0_POCXP_CURRENT_LATCH
18	0_POCXP_CURRENT_LATCH
19	0_POCXP_CURRENT_LATCH
20	0_POCXP_CURRENT_LATCH
21	0_POCXP_CURRENT_LATCH
22	0_POCXP_CURRENT_LATCH
23	0_POCXP_CURRENT_LATCH
24	0_POCXP_CURRENT
25	0_POCXP_CURRENT
26	0_POCXP_CURRENT
27	0_POCXP_CURRENT
28	0_POCXP_CURRENT
29	0_POCXP_CURRENT
30	0_POCXP_CURRENT
31	0_POCXP_CURRENT

0_POCXP_EN_POWER RO, CON104[0], Aon-CXP, Claxon-CXP, Cyton-CXP

CXP Power enabled.

0_POCXP_EN_24V_REG RO, CON104[1], Aon-CXP, Claxon-CXP, Cyton-CXP

24V Regulator enabled.

0_POCXP_EN_CAM_SENSE RO, CON104[2], Aon-CXP, Claxon-CXP, Cyton-CXP

Enable POCXP Sense.

0_POCXP_CAM_IS_POCXP RO, CON104[3], Aon-CXP, Claxon-CXP, Cyton-CXP

Reserved for future use.

0_POCXP_SHORT_DETECTED RO, CON104[4], Aon-CXP, Claxon-CXP, Cyton-CXP

Only valid if POCXP_EN_CAM_SENSE is asserted. This status bit tells if a short circuit was detected from the camera in response to POCXP_EN_CAM_SENSE.

0_POCXP_OPEN_DETECTED RO, CON104[5], Aon-CXP, Claxon-CXP, Cyton-CXP

Only valid if POCXP_EN_CAM_SENSE is asserted. This status bit tells if an open circuit was detected from the camera in response to POCXP_EN_CAM_SENSE. It likely means there is no camera detected.

0_POCXP_OVER_DETECTED RO, CON104[6], Aon-CXP, Claxon-CXP, Cyton-CXP

This status bit indicated over current detected. The protection circuit will immediately disable POCXP_EN_POWER and POCXP_EN_24V_REG in this event. The error is also latched in POCXP_OVER_LATCH.

0_POCXP_OVER_LATCH RO, CON104[7], Aon-CXP, Claxon-CXP, Cyton-CXP

Latched version of POCXP_OVER_DETECTED. Cleared by hardware when both POCXP_EN_POWER and POCXP_EN_24_REG transition from 0 to 1.

0_POCXP_UNDER_DETECTED RO, CON104[8], Aon-CXP, Claxon-CXP, Cyton-CXP

This status bit indicated under current detected. The protection circuit will immediately disable POCXP_EN_POWER and POCXP_EN_24V_REG in this event. The error is also latched in POCXP_UNDER_LATCH. The most likely cause of this is when a powered camera is disconnected.

0_POCXP_UNDER_LATCH RO, CON104[9], Aon-CXP, Claxon-CXP, Cyton-CXP

Latched version of POCXP_UNDER_DETECTED. Cleared by hardware when both POCXP_EN_POWER and POCXP_EN_24_REG transition from 0 to 1.

0_POCXP_24V_OK RO, CON104[10], Aon-CXP, Claxon-CXP, Cyton-CXP

TBD

0_POCXP_STATE RO, CON104[12], Aon-CXP, Claxon-CXP, Cyton-CXP

TBD

0_POCXP_OVR_AUTO_RESTART RW, CON104[13], Aon-CXP, Claxon-CXP, Cyton-CXP

TBD

0_POCXP_SENSE_BYPASS RW, CON104[14], Aon-CXP, Claxon-CXP, Cyton-CXP

TBD

0_ENABLE_POCXP_SYSTEM RW, CON104[15], Aon-CXP, Claxon-CXP, Cyton-CXP

This bit turns on the entire POCXP system. If this bit is zero, this link will not be powered up. If this bit is 1, then the link will be powered if there is a POCXP camera connected. Normally this bit is set to 1 when the driver starts (i.e. when the OS boots up). Once the bit is set to 1, and links the require power will automatically be powered up.

0_POCXP_CURRENT_LATCH RO, CON104[23..16], Aon-CXP, Claxon-CXP, Cyton-CXP

Latched version of POCXP_CURRENT. It is latched on POCXP_UNDER_DETECT or POCXP_OVER_DETECT. Cleared by hardware when both POCXP_EN_POWER and POCXP_EN_24_REG transition from 0 to 1.

0_POCXP_ CURRENT

RO, CON104[31..24], Aon-CXP, Claxon-CXP, Cyton-CXP

Real time current indicator. The scaling is N/255 Amps, where N is the reading of this register in decimal. For 250mA, for example, this register will read ~64. For 32mA it will read ~8 decimal. For currents above 100mA the accuracy is ~1%. For small currents below 50mA the accuracy is ~5%.

10.3 CON105

Bit	Name
0	0_POCXP_OVER_TIMER
1	0_POCXP_OVER_TIMER
2	0_POCXP_OVER_TIMER
3	0_POCXP_OVER_TIMER
4	0_POCXP_OVER_TIMER
5	0_POCXP_OVER_TIMER
6	0_POCXP_OVER_TIMER
7	0_POCXP_OVER_TIMER
8	0_POCXP_OVER_TIMER
9	0_POCXP_OVER_TIMER
10	0_POCXP_OVER_TIMER
11	0_POCXP_OVER_TIMER
12	0_POCXP_OVER_TIMER
13	0_POCXP_OVER_TIMER
14	0_POCXP_OVER_TIMER
15	0_POCXP_OVER_TIMER
16	0_POCXP_OVER_TIMER
17	0_POCXP_OVER_TIMER
18	0_POCXP_OVER_TIMER
19	0_POCXP_OVER_TIMER
20	0_POCXP_OVER_TIMER
21	0_POCXP_OVER_TIMER
22	0_POCXP_OVER_TIMER
23	0_POCXP_OVER_TIMER
24	0_POCXP_OVER_TIMER
25	0_POCXP_OVER_TIMER
26	0_POCXP_OVER_TIMER
27	0_POCXP_OVER_TIMER
28	0_POCXP_OVER_TIMER
29	0_POCXP_OVER_TIMER
30	0_POCXP_OVER_TIMER
31	0_POCXP_OVER_TIMER

0_POCXP_OVER_TIMER

R/W, CON105[31..0], Aon-CXP, Claxon-CXP, Cyton-CXP

This register specifies the time (in 6.4ns units) to wait after both POCXP_EN_POWER and POCXP_EN_24_REG transition from 0 to 1 before enabling the overcurrent detection circuit. It should be set high enough to ignore transients that may occur on initial power.

10.4 CON106

Bit	Name
0	0_POCXP_UNDER_TIMER
1	0_POCXP_UNDER_TIMER
2	0_POCXP_UNDER_TIMER
3	0_POCXP_UNDER_TIMER
4	0_POCXP_UNDER_TIMER
5	0_POCXP_UNDER_TIMER
6	0_POCXP_UNDER_TIMER
7	0_POCXP_UNDER_TIMER
8	0_POCXP_UNDER_TIMER
9	0_POCXP_UNDER_TIMER
10	0_POCXP_UNDER_TIMER
11	0_POCXP_UNDER_TIMER
12	0_POCXP_UNDER_TIMER
13	0_POCXP_UNDER_TIMER
14	0_POCXP_UNDER_TIMER
15	0_POCXP_UNDER_TIMER
16	0_POCXP_UNDER_TIMER
17	0_POCXP_UNDER_TIMER
18	0_POCXP_UNDER_TIMER
19	0_POCXP_UNDER_TIMER
20	0_POCXP_UNDER_TIMER
21	0_POCXP_UNDER_TIMER
22	0_POCXP_UNDER_TIMER
23	0_POCXP_UNDER_TIMER
24	0_POCXP_UNDER_TIMER
25	0_POCXP_UNDER_TIMER
26	0_POCXP_UNDER_TIMER
27	0_POCXP_UNDER_TIMER
28	0_POCXP_UNDER_TIMER
29	0_POCXP_UNDER_TIMER
30	0_POCXP_UNDER_TIMER
31	0_POCXP_UNDER_TIMER

0_POCXP_ UNDER_TIMER

R/W, CON106[31..0], Aon-CXP, Claxon-CXP, Cyton-CXP

This register specifies the time (in 6.4ns units) to wait after both POCXP_EN_POWER and POCXP_EN_24_REG transition from 0 to 1 before enabling the under current detection circuit. It should be set high enough to ignore transients that may occur on initial power.

10.5 CON107

Bit	Name
0	0_COM_RCV_FIFO_SIZE
1	0_COM_RCV_FIFO_SIZE
2	0_COM_RCV_FIFO_SIZE
3	0_COM_RCV_FIFO_SIZE
4	0_COM_RCV_FIFO_SIZE
5	0_COM_RCV_FIFO_SIZE
6	0_COM_RCV_FIFO_SIZE
7	0_COM_RCV_FIFO_SIZE
8	0_COM_RCV_FIFO_SIZE
9	0_COM_RCV_FIFO_SIZE
10	0_COM_RCV_FIFO_SIZE
11	0_COM_RCV_FIFO_SIZE
12	0_COM_RCV_FIFO_SIZE
13	0_COM_RCV_FIFO_SIZE
14	0_COM_RCV_FIFO_SIZE
15	0_COM_RCV_FIFO_SIZE
16	0_COM_SEND_FIFO_SIZE
17	0_COM_SEND_FIFO_SIZE
18	0_COM_SEND_FIFO_SIZE
19	0_COM_SEND_FIFO_SIZE
20	0_COM_SEND_FIFO_SIZE
21	0_COM_SEND_FIFO_SIZE
22	0_COM_SEND_FIFO_SIZE
23	0_COM_SEND_FIFO_SIZE
24	0_COM_SEND_FIFO_SIZE
25	0_COM_SEND_FIFO_SIZE
26	0_COM_SEND_FIFO_SIZE
27	0_COM_SEND_FIFO_SIZE
28	0_COM_SEND_FIFO_SIZE
29	0_COM_SEND_FIFO_SIZE
30	0_COM_SEND_FIFO_SIZE
31	0_COM_SEND_FIFO_SIZE

0_COM_RCV_FIFO_SIZE RO, CON107[15..0], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Depth of the control channel receive FIFO. It is measured in 32-bit words.

0_COM_SEND_FIFO_SIZE RO, CON107[31..16], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Depth of the control channel request FIFO. It is measured in 32-bit words.

10.6 CON108

Bit	Name
0	0_COM_SEND_FIFO_CLR
1	0_COM_SEND_GO
2	Reserved
3	0_COM_CLR_SENT_CNT
4	0_COM_SENT_CNT
5	0_COM_SENT_CNT
6	0_COM_SENT_CNT
7	0_COM_SENT_CNT
8	0_COM_SENT_CNT
9	0_COM_SENT_CNT
10	0_COM_SENT_CNT
11	0_COM_SENT_CNT
12	0_COM_UP_SPD_MANUAL
13	0_COM_UP_SPD
14	Reserved
15	Reserved
16	0_COM_SEND_FIFO_CNT
17	0_COM_SEND_FIFO_CNT
18	0_COM_SEND_FIFO_CNT
19	0_COM_SEND_FIFO_CNT
20	0_COM_SEND_FIFO_CNT
21	0_COM_SEND_FIFO_CNT
22	0_COM_SEND_FIFO_CNT
23	0_COM_SEND_FIFO_CNT
24	0_COM_SEND_FIFO_CNT
25	0_COM_SEND_FIFO_CNT
26	0_COM_SEND_FIFO_CNT
27	0_COM_SEND_FIFO_CNT
28	0_COM_SEND_FIFO_CNT
29	0_COM_SEND_FIFO_CNT
30	0_COM_SEND_FIFO_CNT
31	0_COM_SEND_FIFO_CNT

0_COM_SEND_FIFO_CLR WO, CON108[0], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Clear the control channel request FIFO.

0_COM_SEND_GO WO, CON108[1], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Transmit the packet stored in the request FIFO to the uplink channel. This involves an 8b/10b encoding and serialization. It is assumed that the packet was constructed in the FIFO by software and is ready for transmission.

0_COM_CLR_SENT_CNT WO, CON108[3], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Resets 0_COM_SENT_CNT to zero.

0_COM_SENT_CNT RO, CON108[11..4], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of CXP control packets sent. Poke 0_COM_SENT_CNT to 1 to reset.

0_COM_UP_SPD_MANUAL R/W, CON108[12], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Setting this bit 1 turns off automatic control of the uplink speed. When this bit is 1, the uplink speed is controlled by 0_COM_UP_SPD.

0_COM_UP_SPD R/W, CON108[13], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

When 0_COM_UP_SPD_MANUAL is 1, this bit controls the uplink speed.

0_COM_UP_SPD	Meaning
0	Uplink speed is 20.83 MHz
1	Uplink speed is 41.6 MHz

0_COM_SEND_FIFO_CNT RO, CON108[31..16], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of 32-bit words currently in the request FIFO.

10.7 CON109

Bit	Name
0	0_COM_SEND_DATA
1	0_COM_SEND_DATA
2	0_COM_SEND_DATA
3	0_COM_SEND_DATA
4	0_COM_SEND_DATA
5	0_COM_SEND_DATA
6	0_COM_SEND_DATA
7	0_COM_SEND_DATA
8	0_COM_SEND_DATA
9	0_COM_SEND_DATA
10	0_COM_SEND_DATA
11	0_COM_SEND_DATA
12	0_COM_SEND_DATA
13	0_COM_SEND_DATA
14	0_COM_SEND_DATA
15	0_COM_SEND_DATA
16	0_COM_SEND_DATA
17	0_COM_SEND_DATA
18	0_COM_SEND_DATA
19	0_COM_SEND_DATA
20	0_COM_SEND_DATA
21	0_COM_SEND_DATA
22	0_COM_SEND_DATA
23	0_COM_SEND_DATA
24	0_COM_SEND_DATA
25	0_COM_SEND_DATA
26	0_COM_SEND_DATA
27	0_COM_SEND_DATA
28	0_COM_SEND_DATA
29	0_COM_SEND_DATA
30	0_COM_SEND_DATA
31	0_COM_SEND_DATA

0_COM_SEND_DATA R/W, CON109[31..0], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Software constructs an uplink control channel request packet by writing to the tail of this FIFO as 32-bit words. Software is responsible for constructing the majority of the CXP packed according to the CXP specification. Hardware will automatically generate the leading start of packet indication (K27.7), the control command indication (0x2), and the trailing end of packet indication (K29.7). All other fields (cmd, size, addr, data, crc) are left to software. Once the packet is formed, software should assert the COM_SEND_GO bit. Reading this register returns the last value written.

10.8 CON110

Bit	Name
0	0_COM_RCV_FIFO_CLR
1	Reserved
2	Reserved
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	0_COM_RCV_FIFO_CNT
17	0_COM_RCV_FIFO_CNT
18	0_COM_RCV_FIFO_CNT
19	0_COM_RCV_FIFO_CNT
20	0_COM_RCV_FIFO_CNT
21	0_COM_RCV_FIFO_CNT
22	0_COM_RCV_FIFO_CNT
23	0_COM_RCV_FIFO_CNT
24	0_COM_RCV_FIFO_CNT
25	0_COM_RCV_FIFO_CNT
26	0_COM_RCV_FIFO_CNT
27	0_COM_RCV_FIFO_CNT
28	0_COM_RCV_FIFO_CNT
29	0_COM_RCV_FIFO_CNT
30	0_COM_RCV_FIFO_CNT
31	0_COM_RCV_FIFO_CNT

0_COM_RCV_FIFO_CLR WO, CON110[0], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Clear the control channel receive FIFO.

0_COM_RCV_FIFO_CNT RO, CON110[31..16], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of 32-bit words currently in the response FIFO.

10.9 CON111

Bit	Name
0	0_COM_RCV_DATA
1	0_COM_RCV_DATA
2	0_COM_RCV_DATA
3	0_COM_RCV_DATA
4	0_COM_RCV_DATA
5	0_COM_RCV_DATA
6	0_COM_RCV_DATA
7	0_COM_RCV_DATA
8	0_COM_RCV_DATA
9	0_COM_RCV_DATA
10	0_COM_RCV_DATA
11	0_COM_RCV_DATA
12	0_COM_RCV_DATA
13	0_COM_RCV_DATA
14	0_COM_RCV_DATA
15	0_COM_RCV_DATA
16	0_COM_RCV_DATA
17	0_COM_RCV_DATA
18	0_COM_RCV_DATA
19	0_COM_RCV_DATA
20	0_COM_RCV_DATA
21	0_COM_RCV_DATA
22	0_COM_RCV_DATA
23	0_COM_RCV_DATA
24	0_COM_RCV_DATA
25	0_COM_RCV_DATA
26	0_COM_RCV_DATA
27	0_COM_RCV_DATA
28	0_COM_RCV_DATA
29	0_COM_RCV_DATA
30	0_COM_RCV_DATA
31	0_COM_RCV_DATA

0_COM_RCV_DATA RO, CON111[31..0], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Read the head of the control channel response FIFO. Fifo level can be monitored with COM_RCV_FIFO_CNT.

10.10 CON115

Bit	Name
0	0_LINK_INT_DEST
1	0_LINK_INT_DEST
2	Reserved
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

0_LINK_INT_DEST R/W, CON115[1..0], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

TBD

10.11 CON116

Bit	Name
0	0_UNDER_CURRENT
1	0_OVER_CURRENT
2	0_TRIG_ACK_RCVD
3	0_CTL_ACK_VER_ERR
4	0_CTL_ACK_RCVD
5	0_EVENT_RCVD
6	0_EVENT_FIFO_OVERFLOW
7	0_CTL_RSP_FIFO_OVF
8	0_CTL_REQ_FIFO_OVF
9	0_DOWN_TRIG_RCVD
10	0_TRIG_NOMATCH
11	0_IOACK_UNKNOWN_TYPE
12	0_IOACK_NOMATCH
13	0_HB_ERROR
14	0_HB_RCVD
15	0_STRM_PKT_DROP
16	0_STRM_NOT_ENOUGH_DAT
17	0_STRM_TOO_MUCH_DAT
18	0_STRM_BAD_CRC
19	0_STRM_OVERFLOW
20	0_STRM_CORNER
21	0_SERDES_LOST_ALIGN
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

0_UNDER_CURRENT R/W, CON116[0], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Undercurrent detected by POCXP controller interrupt.

0_OVER_CURRENT R/W, CON116[1], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Overcurrent detected by POCXP controller interrupt.

0_TRIG_ACK_RCVD R/W, CON116[2], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Trigger acknowledgment received from device interrupt.

0_CTL_ACK_VER_ERR R/W, CON116[3], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

There is a version mismatch between an incoming control packet acknowledgment and the board's current version settings. For example, the board could be in CXP 2.0 mode and it receives a CXP 1.1 acknowledgment, then this interrupt would be set.

0_CTL_ACK_RCVD R/W, CON116[4], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Control packet acknowledgment received from device interrupt.

0_EVENT_RCVD R/W, CON116[5], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

CXP 2.0 Event has been received.

0_EVENT_FIFO_OVERFLOW R/W, CON116[6], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

The CXP 2.0 Event FIFO has overflowed.

0_CTL_RSP_FIFO_OVF R/W, CON116[7], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Overflow detected in the control response FIFO (Device to host direction) interrupt.

0_CTL_REQ_FIFO_OVF R/W, CON116[8], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Overflow detected in control request FIFO (Host to device direction) interrupt.

0_DOWN_TRIG_RCVD R/W, CON116[9], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Received CXP trigger from camera.

0_TRIG_NOMATCH R/W, CON116[10], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Problem decoding a trigger packet from device interrupt.

0_IOACK_UNKNOWN_TYPE R/W, CON116[11], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Problem decoding an IO acknowledgment from device interrupt.

0_IOACK_NOMATCH R/W, CON116[12], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Problem decoding an IO acknowledgment from device interrupt.

0_HB_ERROR R/W, CON116[13], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

There is an error decoding a CXP 2.0 Heart Beat Packet.

0_HB_RCVD R/W, CON116[14], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

A CXP 2.0 Heart Beat packet has been received.

0_STRM_PKT_DROP R/W, CON116[15], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Problem decoding a stream packet header, remainder of packet is dropped interrupt.

0_STRM_NOT_ENOUGH_DAT R/W, CON116[16], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Not implemented, reserved interrupt.

0_STRM_TOO_MUCH_DAT R/W, CON116[17], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Not implemented, reserved interrupt.

0_STRM_BAD_CRC R/W, CON116[18], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

CRC error detected in stream packet interrupt.

0_STRM_OVERFLOW R/W, CON116[19], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Overflow in the stream packet buffer interrupt.

0_STRM_CORNER R/W, CON116[19], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Not implemented, reserved interrupt.

0_SERDES_LOST_ALIGN R/W, CON116[21], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Serdes lost alignment interrupt.

10.12 CON117

Bit	Name
0	0_UNDER_CURRENT_M
1	0_OVER_CURRENT_M
2	0_TRIG_ACK_RCVD_M
3	0_GPIO_ACK_RCVD_M
4	0_CTL_ACK_RCVD_M
5	0_GPIO_RCVD_M
6	0_TRIG_RCVD_M
7	0_CTL_RSP_FIFO_OVF_M
8	0_CTL_REQ_FIFO_OVF_M
9	0_DOWN_TRIG_RCVD_M
10	0_TRIG_NOMATCH_M
11	0_IOACK_UNKNOWN_TYPE_M
12	0_IOACK_NOMATCH_M
13	0_IOACK_UNEXPECTED_INT_M
14	0_IOACK_NOMATCH2_M
15	0_STRM_PKT_DROP_M
16	0_STRM_NOT_ENOUGH_DAT_M
17	0_STRM_TOO_MUCH_DAT_M
18	0_STRM_BAD_CRC_M
19	0_STRM_OVERFLOW_M
20	0_STRM_CORNER_M
21	0_SERDES_LOST_ALIGN_M
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

0_UNDER_CURRENT_M R/W, CON117[0], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Mask for the 0_UNDER_CURRENT_M interrupt.

0_OVER_CURRENT_M R/W, CON117[1], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Mask for the 0_OVER_CURRENT_M interrupt.

0_TRIG_ACK_RCVD_M R/W, CON117[2], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Mask for the 0_TRIG_ACK_RCVD_M interrupt.

0_GPIO_ACK_RCVD_M R/W, CON117[3], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Mask for the 0_GPIO_ACK_RCVD_M interrupt.

0_CTL_ACK_RCVD_M R/W, CON117[4], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Mask for the 0_CTL_ACK_RCVD_M interrupt.

0_GPIO_RCVD_M R/W, CON117[5], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Mask for the 0_GPIO_RCVD_M interrupt.

0_TRIG_RCVD_M R/W, CON117[6], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Mask for the 0_TRIG_RCVD_M interrupt.

0_CTL_RSP_FIFO_OVF_M R/W, CON117[7], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Mask for the 0_CTL_RSP_FIFO_OVF_M interrupt.

0_CTL_REQ_FIFO_OVF_M R/W, CON117[8], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Mask for the 0_CTL_REQ_FIFO_OVF_M interrupt.

0_DOWN_TRIG_RCVD_M R/W, CON117[9], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Mask for the 0_DOWN_TRIG_RCVD_M interrupt.

0_TRIG_NOMATCH_M R/W, CON117[10], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Mask for the 0_TRIG_NOMATCH_M interrupt.

0_IOACK_UNKNOWN_TYPE_M R/W, CON117[11], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Mask for the 0_IOACK_UNKNOWN_TYPE_M interrupt.

0_IOACK_NOMATCH_M R/W, CON117[12], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Mask for the 0_IOACK_NOMATCH_M interrupt.

0_IOACK_UNEXPECTED_INT_M R/W, CON117[13], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Mask for the 0_IOACK_UNEXPECTED_INT_M interrupt.

0_IOACK_NOMATCH2_M R/W, CON117[14], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Mask for the 0_IOACK_NOMATCH2_M interrupt.

0_STRM_PKT_DROP_M R/W, CON117[15], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Mask for the 0_STRM_PKT_DROP_M interrupt.

0_STRM_NOT_ENOUGH_DAT_M R/W, CON117[16], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Mask for the 0_STRM_NOT_ENOUGH_DAT_M interrupt.

0_STRM_TOO_MUCH_DAT_M R/W, CON117[17], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Mask for the 0_STRM_TOO_MUCH_DAT_Mv

0_STRM_BAD_CRC_M R/W, CON117[18], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Mask for the 0_STRM_BAD_CRC_M interrupt.

0_STRM_OVERFLOW_M R/W, CON117[19], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Mask for the 0_STRM_OVERFLOW_M interrupt.

10.13 CON118

Bit	Name
0	0_UNDER_CURRENT_WP
1	0_OVER_CURRENT_WP
2	0_TRIG_ACK_RCVD_WP
3	0_GPIO_ACK_RCVD_WP
4	0_CTL_ACK_RCVD_WP
5	0_GPIO_RCVD_WP
6	0_TRIG_RCVD_WP
7	0_CTL_RSP_FIFO_OVF_WP
8	0_CTL_REQ_FIFO_OVF_WP
9	0_DOWN_TRIG_RCVD_WP
10	0_TRIG_NOMATCH_WP
11	0_IOACK_UNKNOWN_TYPE_WP
12	0_IOACK_NOMATCH_WP
13	0_IOACK_UNEXPECTED_INT_WP
14	0_IOACK_NOMATCH2_WP
15	0_STRM_PKT_DROP_WP
16	0_STRM_NOT_ENOUGH_DAT_WP
17	0_STRM_TOO_WPUCH_DAT_WP
18	0_STRM_BAD_CRC_WP
19	0_STRM_OVERFLOW_WP
20	0_STRM_CORNER_WP
21	0_SERDES_LOST_ALIGN_WP
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

0_UNDER_CURRENT_WP R/W, CON118[0], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Write mask for the 0_UNDER_CURRENT_WP interrupt.

0_OVER_CURRENT_WP R/W, CON118[1], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Write mask for the 0_OVER_CURRENT_WP interrupt.

0_TRIG_ACK_RCVD_WP R/W, CON118[2], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Write mask for the 0_TRIG_ACK_RCVD_WP interrupt.

0_GPIO_ACK_RCVD_WP R/W, CON118[3], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Write mask for the 0_GPIO_ACK_RCVD_WP interrupt.

0_CTL_ACK_RCVD_WP R/W, CON118[4], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Write mask for the 0_CTL_ACK_RCVD_WP interrupt.

0_GPIO_RCVD_WP R/W, CON118[5], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Write mask for the 0_GPIO_RCVD_WP interrupt.

0_TRIG_RCVD_WP R/W, CON118[6], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Write mask for the 0_TRIG_RCVD_WP interrupt.

0_CTL_RSP_FIFO_OVF_WP R/W, CON118[7], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Write mask for the 0_CTL_RSP_FIFO_OVF_WP interrupt.

0_CTL_REQ_FIFO_OVF_WP R/W, CON118[8], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Write mask for the 0_CTL_REQ_FIFO_OVF_WP interrupt.

0_DOWN_TRIG_RCVD_WP R/W, CON118[9], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Write mask for the 0_DOWN_TRIG_RCVD_WP interrupt.

0_TRIG_NOMATCH_WP R/W, CON118[10], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Write mask for the 0_TRIG_NOMATCH_WP interrupt.

0_IOACK_UNKNOWN_TYPE_WP R/W, CON118[11], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Write mask for the 0_IOACK_UNKNOWN_TYPE_WP interrupt.

0_IOACK_NOMATCH_WP R/W, CON118[12], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Write mask for the 0_IOACK_NOMATCH_WP interrupt.

0_IOACK_UNEXPECTED_INT_WP R/W, CON118[13], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Write mask for the 0_IOACK_UNEXPECTED_INT_WP interrupt.

0_IOACK_NOMATCH2_WP R/W, CON118[14], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Write mask for the 0_IOACK_NOMATCH2_WP interrupt.

0_STRM_PKT_DROP_WP R/W, CON118[15], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Write mask for the 0_STRM_PKT_DROP_WP interrupt.

0_STRM_NOT_ENOUGH_DAT_WP R/W, CON118[16], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Write mask for the 0_STRM_NOT_ENOUGH_DAT_WP interrupt.

0_STRM_TOO_WPUCH_DAT_WP R/W, CON118[17], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Write mask for the 0_STRM_TOO_WPUCH_DAT_WP interrupt.

0_STRM_BAD_CRC_WP R/W, CON118[18], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Write mask for the 0_STRM_BAD_CRC_WP interrupt.

0_STRM_OVERFLOW_WP R/W, CON118[19], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Write mask for the 0_STRM_OVERFLOW_WP interrupt.

10.14 CON120

Bit	Name
0	0_PKT_RCVD_CNT
1	0_PKT_RCVD_CNT
2	0_PKT_RCVD_CNT
3	0_PKT_RCVD_CNT
4	0_PKT_RCVD_CNT
5	0_PKT_RCVD_CNT
6	0_PKT_RCVD_CNT
7	0_PKT_RCVD_CNT
8	0_PKT_RCVD_CNT
9	0_PKT_RCVD_CNT
10	0_PKT_RCVD_CNT
11	0_PKT_RCVD_CNT
12	0_PKT_RCVD_CNT
13	0_PKT_RCVD_CNT
14	0_PKT_RCVD_CNT
15	0_PKT_RCVD_CNT
16	0_PKT_GNT_CNT
17	0_PKT_GNT_CNT
18	0_PKT_GNT_CNT
19	0_PKT_GNT_CNT
20	0_PKT_GNT_CNT
21	0_PKT_GNT_CNT
22	0_PKT_GNT_CNT
23	0_PKT_GNT_CNT
24	0_PKT_GNT_CNT
25	0_PKT_GNT_CNT
26	0_PKT_GNT_CNT
27	0_PKT_GNT_CNT
28	0_PKT_GNT_CNT
29	0_PKT_GNT_CNT
30	0_PKT_GNT_CNT
31	0_PKT_GNT_CNT

0_PKT_RCVD_CNT RO, CON120[15..0], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of CXP packets received on this link. The entire register CON120[31:0] is cleared on a write access of any value.

0_PKT_GNT_CNT RO, CON120[31..16], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of packets forwarded from this link to the Stream Assembler engine. The entire register CON120[31:0] is cleared on a write access of any value.

10.15 CON121

Bit	Name
0	0_PKT_DROP_CNT
1	0_PKT_DROP_CNT
2	0_PKT_DROP_CNT
3	0_PKT_DROP_CNT
4	0_PKT_DROP_CNT
5	0_PKT_DROP_CNT
6	0_PKT_DROP_CNT
7	0_PKT_DROP_CNT
8	0_PKT_DROP_CNT
9	0_PKT_DROP_CNT
10	0_PKT_DROP_CNT
11	0_PKT_DROP_CNT
12	0_PKT_DROP_CNT
13	0_PKT_DROP_CNT
14	0_PKT_DROP_CNT
15	0_PKT_DROP_CNT
16	0_CRC_ERR_CNT
17	0_CRC_ERR_CNT
18	0_CRC_ERR_CNT
19	0_CRC_ERR_CNT
20	0_CRC_ERR_CNT
21	0_CRC_ERR_CNT
22	0_CRC_ERR_CNT
23	0_CRC_ERR_CNT
24	0_CRC_ERR_CNT
25	0_CRC_ERR_CNT
26	0_CRC_ERR_CNT
27	0_CRC_ERR_CNT
28	0_CRC_ERR_CNT
29	0_CRC_ERR_CNT
30	0_CRC_ERR_CNT
31	0_CRC_ERR_CNT

0_PKT_DROP_CNT RO CON121[15..0], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of packets dropped due to packet header errors. The entire register [31:0] is cleared on a write access of any value.

0_CRC_ERR_CNT RO, CON121[31..16], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of packets with crc errors. These packets are not dropped because they most likely have a small data error and the majority of the frame can be recovered. The entire register [31:0] is cleared on a write access of any value.

10.16 CON122

Bit	Name
0	0_CXP_TRIG_FALL_SENT
1	0_CXP_TRIG_FALL_SENT
2	0_CXP_TRIG_FALL_SENT
3	0_CXP_TRIG_FALL_SENT
4	0_CXP_TRIG_FALL_SENT
5	0_CXP_TRIG_FALL_SENT
6	0_CXP_TRIG_FALL_SENT
7	0_CXP_TRIG_FALL_SENT
8	0_CXP_TRIG_RISE_SENT
9	0_CXP_TRIG_RISE_SENT
10	0_CXP_TRIG_RISE_SENT
11	0_CXP_TRIG_RISE_SENT
12	0_CXP_TRIG_RISE_SENT
13	0_CXP_TRIG_RISE_SENT
14	0_CXP_TRIG_RISE_SENT
15	0_CXP_TRIG_RISE_SENT
16	0_CXP_TRIG_ACK_CNT
17	0_CXP_TRIG_ACK_CNT
18	0_CXP_TRIG_ACK_CNT
19	0_CXP_TRIG_ACK_CNT
20	0_CXP_TRIG_ACK_CNT
21	0_CXP_TRIG_ACK_CNT
22	0_CXP_TRIG_ACK_CNT
23	0_CXP_TRIG_ACK_CNT
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	0_CXP_TRIG_STATE

0_CXP_TRIG_FALL_SENT RO, CON122[7..0], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of CXP trigger falling edge trigger packets sent to the camera.

0_CXP_TRIG_RISE_SENT RO, CON122[15..8], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of CXP trigger rising edge trigger packets sent to the camera.

0_CXP_TRIG_ACK_CNT RO, CON122[23..16], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of CXP trigger acknowledgments received from the camera.

0_CXP_TRIG_STATE RO, CON122[31], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Current state of the uplink CXP trigger signal.

10.17 CON123

Bit	Name
0	0_DT_FALL_RCVD_COUNT
1	0_DT_FALL_RCVD_COUNT
2	0_DT_FALL_RCVD_COUNT
3	0_DT_FALL_RCVD_COUNT
4	0_DT_FALL_RCVD_COUNT
5	0_DT_FALL_RCVD_COUNT
6	0_DT_FALL_RCVD_COUNT
7	0_DT_FALL_RCVD_COUNT
8	0_DT_RISE_RCVD_COUNT
9	0_DT_RISE_RCVD_COUNT
10	0_DT_RISE_RCVD_COUNT
11	0_DT_RISE_RCVD_COUNT
12	0_DT_RISE_RCVD_COUNT
13	0_DT_RISE_RCVD_COUNT
14	0_DT_RISE_RCVD_COUNT
15	0_DT_RISE_RCVD_COUNT
16	0_DT_TRIG_DELAY
17	0_DT_TRIG_DELAY
18	0_DT_TRIG_DELAY
19	0_DT_TRIG_DELAY
20	0_DT_TRIG_DELAY
21	0_DT_TRIG_DELAY
22	0_DT_TRIG_DELAY
23	0_DT_TRIG_DELAY
24	0_DT_FALL_DISABLE
25	0_DT_RISE_DISABLE
26	0_DT_TRIG_NUM
27	0_DT_TRIG_NUM
28	0_DT_TRIG_NUM
29	0_DT_TRIG_NUM
30	0_DT_CLR_CNT
31	0_DT_STATE

0_DT_FALL_RCVD_COUNT RO, CON123[7..0], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of CXP down stream (camera to frame grabber) trigger falling edges.

0_DT_RISE_RCVD_COUNT RO, CON123[15..8], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of CXP down stream (camera to frame grabber) trigger rising edges.

0_DT_TRIG_DELAY RO, CON123[23..16], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

The payload of the CXP down stream (camera to frame grabber) trigger.

0_DT_FALL_DISABLE R/W, CON123[24], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Setting this bit to 1 will disable the falling edge trigger packet from causing an interrupt. It will still be counted and affect DT_STATE.

0_DT_RISE_DISABLE R/W, CON123[25], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Setting this bit to 1 will disable the rising edge trigger packet from causing an interrupt. It will still be counted and affect DT_STATE.

0_DT_TRIG_NUM RO, CON123[29..26], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

When the link is in CXP 2.0 or later mode, these bits capture the TriggerN field of a trigger.

0_DT_CLR_CNT WO, CON123[30], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Writing this register to 1 will clear that counters in this registers.

0_DT_STATE RO, CON123[31], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Current stated of the downstream trigger.

10.18 CON127

Bit	Name
0	0_SERDES_STATE
1	0_SERDES_STATE
2	0_SERDES_STATE
3	0_SERDES_STATE
4	0_SERDES_STATE
5	0_SERDES_STATE
6	0_SERDES_STATE
7	0_SERDES_STATE
8	0_LINK_SPEED
9	0_LINK_SPEED
10	0_LINK_SPEED
11	0_LINK_SPEED
12	0_LINK_SPEED
13	0_LINK_SPEED
14	0_LINK_SPEED
15	0_LINK_SPEED
16	0_LOST_ALIGN_CNT
17	0_LOST_ALIGN_CNT
18	0_LOST_ALIGN_CNT
19	0_LOST_ALIGN_CNT
20	0_LOST_ALIGN_CNT
21	0_LOST_ALIGN_CNT
22	0_LOST_ALIGN_CNT
23	0_LOST_ALIGN_CNT
24	0_LOST_ALIGN_CNT
25	0_LOST_ALIGN_CNT
26	Reserved
27	Reserved
28	0_RESYNC_TOG_MSTR
29	0_RESYNC_TOG
30	0_SERDES_ALIGNED
31	Reserved

0_SERDES_STATE RO, CON127[7..0], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Current state of the SERDES associated with this link.

0_LINK_SPEED RO, CON127[15..8], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Current speed of the SERDES associated with this link.

0_LOST_ALIGN_CNT RO, CON127[25..16], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of times the SERDES associated with this link has lost alignment.

0_RESYNC_TOG_MSTR RO, CON127[28], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Diagnostics.

0_RESYNC_TOG RO, CON127[29], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Diagnostics.

0_SERDES_ALIGNED RO, CON127[30], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

If this bit is 1 the SERDES associated with this link is aligned.

10.19 CON128

Bit	Name
0	0_RAW_DATA_MODE
1	0_REMOVE_IDLES
2	0_DISABLE_SUB
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	0_MAX_PKT_RCVD
17	0_MAX_PKT_RCVD
18	0_MAX_PKT_RCVD
19	0_MAX_PKT_RCVD
20	0_MAX_PKT_RCVD
21	0_MAX_PKT_RCVD
22	0_MAX_PKT_RCVD
23	0_MAX_PKT_RCVD
24	0_MAX_PKT_RCVD
25	0_MAX_PKT_RCVD
26	0_MAX_PKT_RCVD
27	0_MAX_PKT_RCVD
28	0_MAX_PKT_RCVD
29	0_MAX_PKT_RCVD
30	0_MAX_PKT_RCVD
31	0_MAX_PKT_RCVD

0_RAW_DATA_MODE R/W, CON128[0], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Enable raw capture mode. This mode captures raw data from the CXP link without any processing.

0_REMOVE_IDLES R/W, CON128[1], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Remove CXP idle packets from the raw capture.

0_DISABLE_SUB R/W, CON128[2], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the automatic pixel substitution facility built into the board's receiver circuit. In some very unusual circumstance that low level CXP hardware and perform a pixel substitution if bad data packet is detected. This substitution will still result in a CRC error for the packet, but it will cause less harm than if the original bad packet was received. Setting this bit to 1 will turn this facility off, which is really only useful for testing.

0_MAX_PKT_RCVD R/W, CON128[16], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

The maximum CXP packet that has been received from the camera. Units are 32-bit words. Clear this register by writing a 0 to it.

10.20 CON131

Bit	Name
0	0_SERDES_ERROR_CODE
1	0_SERDES_ERROR_CODE
2	0_SERDES_ERROR_CODE
3	0_SERDES_ERROR_CODE
4	0_SERDES_ERROR_CODE
5	0_SERDES_ERROR_CODE
6	0_SERDES_ERROR_CODE
7	0_SERDES_ERROR_CODE
8	0_SERDES_ERROR_CODE
9	0_SERDES_ERROR_CODE
10	0_SERDES_ERROR_CODE
11	0_SERDES_ERROR_CODE
12	0_SERDES_ERROR_CODE
13	0_SERDES_ERROR_CODE
14	0_SERDES_ERROR_CODE
15	0_SERDES_ERROR_CODE
16	0_SERDES_ERROR_CODE
17	0_SERDES_ERROR_CODE
18	0_SERDES_ERROR_CODE
19	0_SERDES_ERROR_CODE
20	0_SERDES_ERROR_CODE
21	0_SERDES_ERROR_CODE
22	0_SERDES_ERROR_CODE
23	0_SERDES_ERROR_CODE
24	0_SERDES_ERROR_CODE
25	0_SERDES_ERROR_CODE
26	0_SERDES_ERROR_CODE
27	0_SERDES_ERROR_CODE
28	0_SERDES_ERROR_CODE
29	0_SERDES_ERROR_CODE
30	0_SERDES_ERROR_CODE
31	Reserved

0_SERDES_ERROR_CODE RO, CON131[30..0], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Describe 0_SERDES_ERROR_CODE

10.21 CON136

Bit	Name
0	1_POCXP_EN_POWER
1	1_POCXP_EN_24V_REG
2	1_POCXP_EN_CAM_SENSE
3	1_POCXP_CAM_IS_POCXP
4	1_POCXP_SHORT_DETECTED
5	1_POCXP_OPEN_DETECTED
6	1_POCXP_OVER_DETECTED
7	1_POCXP_OVER_LATCH
8	1_POCXP_UNDER_DETECTED
9	1_POCXP_UNDER_LATCH
10	1_POCXP_24V_OK
11	Reserved
12	1_POCXP_STATE
13	1_POCXP_OVR_AUTO_RESTART
14	1_POCXP_SENSE_BYPASS
15	1_ENABLE_POCXP_SYSTEM
16	1_POCXP_CURRENT_LATCH
17	1_POCXP_CURRENT_LATCH
18	1_POCXP_CURRENT_LATCH
19	1_POCXP_CURRENT_LATCH
20	1_POCXP_CURRENT_LATCH
21	1_POCXP_CURRENT_LATCH
22	1_POCXP_CURRENT_LATCH
23	1_POCXP_CURRENT_LATCH
24	1_POCXP_CURRENT
25	1_POCXP_CURRENT
26	1_POCXP_CURRENT
27	1_POCXP_CURRENT
28	1_POCXP_CURRENT
29	1_POCXP_CURRENT
30	1_POCXP_CURRENT
31	1_POCXP_CURRENT

1_POCXP_EN_POWER RO, CON136[0], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_EN_POWER

1_POCXP_EN_24V_REG RO, CON136[1], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_EN_24V_REG

1_POCXP_EN_CAM_SENSE RO, CON136[2], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_EN_CAM_SENSE

1_POCXP_CAM_IS_POCXP RO, CON136[3], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_CAM_IS_POCXP

1_POCXP_SHORT_DETECTED RO, CON136[4], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_SHORT_DETECTED

1_POCXP_OPEN_DETECTED RO, CON136[5], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_OPEN_DETECTED

1_POCXP_OVER_DETECTED RO, CON136[6], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_OVER_DETECTED

1_POCXP_OVER_LATCH RO, CON136[7], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_OVER_LATCH

1_POCXP_UNDER_DETECTED RO, CON136[8], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_UNDER_DETECTED

1_POCXP_UNDER_LATCH RO, CON136[9], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_UNDER_LATCH

1_POCXP_24V_OK RO, CON136[10], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_24V_OK

1_POCXP_STATE RO, CON136[12], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_STATE.

1_POCXP_OVR_AUTO_RESTART RW, CON136[13], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_OVR_AUTO_RESTART.

1_POCXP_SENSE_BYPASS RW, CON136[14], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_SENSE_BYPASS.

1_ENABLE_POCXP_SYSTEM RW, CON136[15], Claxon-CXP, Cyton-CXP

See description of 0_ENABLE_POCXP_SYSTEM.

1_POCXP_CURRENT_LATCH RO, CON136[23..16], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_CURRENT_LATCH

1_POCXP_CURRENT RO, CON136[31..24], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_CURRENT

10.22 CON137

Bit	Name
0	1_POCXP_OVER_TIMER
1	1_POCXP_OVER_TIMER
2	1_POCXP_OVER_TIMER
3	1_POCXP_OVER_TIMER
4	1_POCXP_OVER_TIMER
5	1_POCXP_OVER_TIMER
6	1_POCXP_OVER_TIMER
7	1_POCXP_OVER_TIMER
8	1_POCXP_OVER_TIMER
9	1_POCXP_OVER_TIMER
10	1_POCXP_OVER_TIMER
11	1_POCXP_OVER_TIMER
12	1_POCXP_OVER_TIMER
13	1_POCXP_OVER_TIMER
14	1_POCXP_OVER_TIMER
15	1_POCXP_OVER_TIMER
16	1_POCXP_OVER_TIMER
17	1_POCXP_OVER_TIMER
18	1_POCXP_OVER_TIMER
19	1_POCXP_OVER_TIMER
20	1_POCXP_OVER_TIMER
21	1_POCXP_OVER_TIMER
22	1_POCXP_OVER_TIMER
23	1_POCXP_OVER_TIMER
24	1_POCXP_OVER_TIMER
25	1_POCXP_OVER_TIMER
26	1_POCXP_OVER_TIMER
27	1_POCXP_OVER_TIMER
28	1_POCXP_OVER_TIMER
29	1_POCXP_OVER_TIMER
30	1_POCXP_OVER_TIMER
31	1_POCXP_OVER_TIMER

1_POCXP_OVER_TIMER R/W, CON137[31..0], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_OVER_TIMER

10.23 CON138

Bit	Name
0	1_POCXP_UNDER_TIMER
1	1_POCXP_UNDER_TIMER
2	1_POCXP_UNDER_TIMER
3	1_POCXP_UNDER_TIMER
4	1_POCXP_UNDER_TIMER
5	1_POCXP_UNDER_TIMER
6	1_POCXP_UNDER_TIMER
7	1_POCXP_UNDER_TIMER
8	1_POCXP_UNDER_TIMER
9	1_POCXP_UNDER_TIMER
10	1_POCXP_UNDER_TIMER
11	1_POCXP_UNDER_TIMER
12	1_POCXP_UNDER_TIMER
13	1_POCXP_UNDER_TIMER
14	1_POCXP_UNDER_TIMER
15	1_POCXP_UNDER_TIMER
16	1_POCXP_UNDER_TIMER
17	1_POCXP_UNDER_TIMER
18	1_POCXP_UNDER_TIMER
19	1_POCXP_UNDER_TIMER
20	1_POCXP_UNDER_TIMER
21	1_POCXP_UNDER_TIMER
22	1_POCXP_UNDER_TIMER
23	1_POCXP_UNDER_TIMER
24	1_POCXP_UNDER_TIMER
25	1_POCXP_UNDER_TIMER
26	1_POCXP_UNDER_TIMER
27	1_POCXP_UNDER_TIMER
28	1_POCXP_UNDER_TIMER
29	1_POCXP_UNDER_TIMER
30	1_POCXP_UNDER_TIMER
31	1_POCXP_UNDER_TIMER

1_POCXP_ UNDER_TIMER R/W, CON138[31..0], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_UNDER_TIMER

10.24 CON139

Bit	Name
0	1_COM_RCV_FIFO_SIZE
1	1_COM_RCV_FIFO_SIZE
2	1_COM_RCV_FIFO_SIZE
3	1_COM_RCV_FIFO_SIZE
4	1_COM_RCV_FIFO_SIZE
5	1_COM_RCV_FIFO_SIZE
6	1_COM_RCV_FIFO_SIZE
7	1_COM_RCV_FIFO_SIZE
8	1_COM_RCV_FIFO_SIZE
9	1_COM_RCV_FIFO_SIZE
10	1_COM_RCV_FIFO_SIZE
11	1_COM_RCV_FIFO_SIZE
12	1_COM_RCV_FIFO_SIZE
13	1_COM_RCV_FIFO_SIZE
14	1_COM_RCV_FIFO_SIZE
15	1_COM_RCV_FIFO_SIZE
16	1_COM_SEND_FIFO_SIZE
17	1_COM_SEND_FIFO_SIZE
18	1_COM_SEND_FIFO_SIZE
19	1_COM_SEND_FIFO_SIZE
20	1_COM_SEND_FIFO_SIZE
21	1_COM_SEND_FIFO_SIZE
22	1_COM_SEND_FIFO_SIZE
23	1_COM_SEND_FIFO_SIZE
24	1_COM_SEND_FIFO_SIZE
25	1_COM_SEND_FIFO_SIZE
26	1_COM_SEND_FIFO_SIZE
27	1_COM_SEND_FIFO_SIZE
28	1_COM_SEND_FIFO_SIZE
29	1_COM_SEND_FIFO_SIZE
30	1_COM_SEND_FIFO_SIZE
31	1_COM_SEND_FIFO_SIZE

1_COM_RCV_FIFO_SIZE RO, CON139[15..0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_RCV_FIFO_SIZE

1_COM_SEND_FIFO_SIZE RO, CON139[31..16], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_SEND_FIFO_SIZE

10.25 CON140

Bit	Name
0	1_COM_SEND_FIFO_CLR
1	1_COM_SEND_GO
2	Reserved
3	1_COM_CLR_SENT_CNT
4	1_COM_SENT_CNT
5	1_COM_SENT_CNT
6	1_COM_SENT_CNT
7	1_COM_SENT_CNT
8	1_COM_SENT_CNT
9	1_COM_SENT_CNT
10	1_COM_SENT_CNT
11	1_COM_SENT_CNT
12	1_COM_UP_SPD_MANUAL
13	1_COM_UP_SPD
14	Reserved
15	Reserved
16	1_COM_SEND_FIFO_CNT
17	1_COM_SEND_FIFO_CNT
18	1_COM_SEND_FIFO_CNT
19	1_COM_SEND_FIFO_CNT
20	1_COM_SEND_FIFO_CNT
21	1_COM_SEND_FIFO_CNT
22	1_COM_SEND_FIFO_CNT
23	1_COM_SEND_FIFO_CNT
24	1_COM_SEND_FIFO_CNT
25	1_COM_SEND_FIFO_CNT
26	1_COM_SEND_FIFO_CNT
27	1_COM_SEND_FIFO_CNT
28	1_COM_SEND_FIFO_CNT
29	1_COM_SEND_FIFO_CNT
30	1_COM_SEND_FIFO_CNT
31	1_COM_SEND_FIFO_CNT

1_COM_SEND_FIFO_CLR WO, CON140[0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_SEND_FIFO_CLR

1_COM_SEND_GO WO, CON140[1], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 1_COM_SEND_GO

1_COM_CLR_SENT_CNT WO, CON140[3], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_CLR_SENT_CNT

1_COM_SENT_CNT RO, CON140[11..4], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_SENT_CNT.

1_COM_UP_SPD_MANUAL R/W, CON140[12], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_UP_SPD_MANUAL.

1_COM_UP_SPD R/W, CON140[13], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_UP_SPD.

1_COM_SEND_FIFO_CNT RO, CON140[31..16], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 1_COM_SEND_FIFO_CNT

10.26 CON141

Bit	Name
0	1_COM_SEND_DATA
1	1_COM_SEND_DATA
2	1_COM_SEND_DATA
3	1_COM_SEND_DATA
4	1_COM_SEND_DATA
5	1_COM_SEND_DATA
6	1_COM_SEND_DATA
7	1_COM_SEND_DATA
8	1_COM_SEND_DATA
9	1_COM_SEND_DATA
10	1_COM_SEND_DATA
11	1_COM_SEND_DATA
12	1_COM_SEND_DATA
13	1_COM_SEND_DATA
14	1_COM_SEND_DATA
15	1_COM_SEND_DATA
16	1_COM_SEND_DATA
17	1_COM_SEND_DATA
18	1_COM_SEND_DATA
19	1_COM_SEND_DATA
20	1_COM_SEND_DATA
21	1_COM_SEND_DATA
22	1_COM_SEND_DATA
23	1_COM_SEND_DATA
24	1_COM_SEND_DATA
25	1_COM_SEND_DATA
26	1_COM_SEND_DATA
27	1_COM_SEND_DATA
28	1_COM_SEND_DATA
29	1_COM_SEND_DATA
30	1_COM_SEND_DATA
31	1_COM_SEND_DATA

1_COM_SEND_DATA R/W, CON141[31..0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_SEND_DATA

10.27 CON142

Bit	Name
0	1_COM_RCV_FIFO_CLR
1	Reserved
2	Reserved
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	1_COM_RCV_FIFO_CNT
17	1_COM_RCV_FIFO_CNT
18	1_COM_RCV_FIFO_CNT
19	1_COM_RCV_FIFO_CNT
20	1_COM_RCV_FIFO_CNT
21	1_COM_RCV_FIFO_CNT
22	1_COM_RCV_FIFO_CNT
23	1_COM_RCV_FIFO_CNT
24	1_COM_RCV_FIFO_CNT
25	1_COM_RCV_FIFO_CNT
26	1_COM_RCV_FIFO_CNT
27	1_COM_RCV_FIFO_CNT
28	1_COM_RCV_FIFO_CNT
29	1_COM_RCV_FIFO_CNT
30	1_COM_RCV_FIFO_CNT
31	1_COM_RCV_FIFO_CNT

1_COM_RCV_FIFO_CLR WO, CON142[0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_RCV_FIFO_CLR

1_COM_RCV_FIFO_CNT RO, CON142[31..16], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_RCV_FIFO_CNT

10.28 CON143

Bit	Name
0	1_COM_RCV_DATA
1	1_COM_RCV_DATA
2	1_COM_RCV_DATA
3	1_COM_RCV_DATA
4	1_COM_RCV_DATA
5	1_COM_RCV_DATA
6	1_COM_RCV_DATA
7	1_COM_RCV_DATA
8	1_COM_RCV_DATA
9	1_COM_RCV_DATA
10	1_COM_RCV_DATA
11	1_COM_RCV_DATA
12	1_COM_RCV_DATA
13	1_COM_RCV_DATA
14	1_COM_RCV_DATA
15	1_COM_RCV_DATA
16	1_COM_RCV_DATA
17	1_COM_RCV_DATA
18	1_COM_RCV_DATA
19	1_COM_RCV_DATA
20	1_COM_RCV_DATA
21	1_COM_RCV_DATA
22	1_COM_RCV_DATA
23	1_COM_RCV_DATA
24	1_COM_RCV_DATA
25	1_COM_RCV_DATA
26	1_COM_RCV_DATA
27	1_COM_RCV_DATA
28	1_COM_RCV_DATA
29	1_COM_RCV_DATA
30	1_COM_RCV_DATA
31	1_COM_RCV_DATA

1_COM_RCV_DATA RO, CON143[31..0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_RCV_DATA

10.29 CON147

Bit	Name
0	1_LINK_INT_DEST
1	1_LINK_INT_DEST
2	Reserved
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

1_LINK_INT_DEST R/W, CON147[1..0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_LINK_INT_DEST

10.30 CON148

Bit	Name
0	1_UNDER_CURRENT
1	1_OVER_CURRENT
2	1_TRIG_ACK_RCVD
3	1_CTL_ACK_VER_ERR
4	1_CTL_ACK_RCVD
5	1_EVENT_RCVD
6	1_EVENT_FIFO_OVERFLOW
7	1_CTL_RSP_FIFO_OVF
8	1_CTL_REQ_FIFO_OVF
9	1_DOWN_TRIG_RCVD
10	1_TRIG_NOMATCH
11	1_IOACK_UNKNOWN_TYPE
12	1_IOACK_NOMATCH
13	1_HB_ERROR
14	1_HB_RCVD
15	1_STRM_PKT_DROP
16	1_STRM_NOT_ENOUGH_DAT
17	1_STRM_TOO_MUCH_DAT
18	1_STRM_BAD_CRC
19	1_STRM_OVERFLOW
20	1_STRM_CORNER
21	1_SERDES_LOST_ALIGN
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

1_UNDER_CURRENT R/W, CON148[0], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Undercurrent detected by POCXP controller interrupt.

1_OVER_CURRENT R/W, CON148[1], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Overcurrent detected by POCXP controller interrupt.

1_TRIG_ACK_RCVD R/W, CON148[2], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Trigger acknowledgment received from device interrupt.

1_CTL_ACK_VER_ERR R/W, CON148[3], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

There is a version mismatch between an incoming control packet acknowledgment and the board's current version settings. For example, the board could be in CXP 2.0 mode and it receives a CXP 1.1 acknowledgment, then this interrupt would be set.

1_CTL_ACK_RCVD R/W, CON148[4], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Control packet acknowledgment received from device interrupt.

1_EVENT_RCVD R/W, CON148[5], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

CXP 2.0 Event has been received.

1_EVENT_FIFO_OVERFLOW R/W, CON148[6], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

The CXP 2.0 Event FIFO has overflowed.

1_CTL_RSP_FIFO_OVF R/W, CON148[7], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Overflow detected in the control response FIFO (Device to host direction) interrupt.

1_CTL_REQ_FIFO_OVF R/W, CON148[8], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Overflow detected in control request FIFO (Host to device direction) interrupt.

1_DOWN_TRIG_RCVD R/W, CON148[9], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Received CXP trigger from camera.

1_TRIG_NOMATCH R/W, CON148[10], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Problem decoding a trigger packet from device interrupt.

1_IOACK_UNKNOWN_TYPE R/W, CON148[11], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Problem decoding an IO acknowledgment from device interrupt.

1_IOACK_NOMATCH R/W, CON148[12], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Problem decoding an IO acknowledgment from device interrupt.

1_HB_ERROR R/W, CON148[13], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

There is an error decoding a CXP 2.0 Heart Beat Packet.

1_HB_RCVD R/W, CON148[14], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

A CXP 2.0 Heart Beat packet has been received.

1_STRM_PKT_DROP R/W, CON148[15], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Problem decoding a stream packet header, remainder of packet is dropped interrupt.

1_STRM_NOT_ENOUGH_DAT R/W, CON148[16], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Not implemented, reserved interrupt.

1_STRM_TOO_MUCH_DAT R/W, CON148[17], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Not implemented, reserved interrupt.

1_STRM_BAD_CRC R/W, CON148[18], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

CRC error detected in stream packet interrupt.

1_STRM_OVERFLOW R/W, CON148[19], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Overflow in the stream packet buffer interrupt.

1_STRM_ CORNER R/W, CON148[19], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Not implemented, reserved interrupt.

1_SERDES_ LOST_ALIGN R/W, CON148[21], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

SERDES lost alignment interrupt.

10.31 CON149

Bit	Name
0	1_UNDER_CURRENT_M
1	1_OVER_CURRENT_M
2	1_TRIG_ACK_RCVD_M
3	1_GPIO_ACK_RCVD_M
4	1_CTL_ACK_RCVD_M
5	1_GPIO_RCVD_M
6	1_TRIG_RCVD_M
7	1_CTL_RSP_FIFO_OVF_M
8	1_CTL_REQ_FIFO_OVF_M
9	1_DOWN_TRIG_RCVD_M
10	1_TRIG_NOMATCH_M
11	1_IOACK_UNKNOWN_TYPE_M
12	1_IOACK_NOMATCH_M
13	1_IOACK_UNEXPECTED_INT_M
14	1_IOACK_NOMATCH2_M
15	1_STRM_PKT_DROP_M
16	1_STRM_NOT_ENOUGH_DAT_M
17	1_STRM_TOO_MUCH_DAT_M
18	1_STRM_BAD_CRC_M
19	1_STRM_OVERFLOW_M
20	1_STRM_CORNER_M
21	1_SERDES_LOST_ALIGN_M
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

1_UNDER_CURRENT_M R/W, CON149[0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_UNDER_CURRENT_M

1_OVER_CURRENT_M R/W, CON149[1], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_OVER_CURRENT_M

1_TRIG_ACK_RCVD_M R/W, CON149[2], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_TRIG_ACK_RCVD_M

1_GPIO_ACK_RCVD_M R/W, CON149[3], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_GPIO_ACK_RCVD_M

1_CTL_ACK_RCVD_M R/W, CON149[4], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_CTL_ACK_RCVD_M

1_GPIO_RCVD_M R/W, CON149[5], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_GPIO_RCVD_M

1_TRIG_RCVD_M R/W, CON149[6], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_TRIG_RCVD_M

1_CTL_RSP_FIFO_OVF_M R/W, CON149[7], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_CTL_RSP_FIFO_OVF_M

1_CTL_REQ_FIFO_OVF_M R/W, CON149[8], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_CTL_REQ_FIFO_OVF_M

1_DOWN_TRIG_RCVD_M R/W, CON149[9], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_DOWN_TRIG_RCVD_M

1_TRIG_NOMATCH_M R/W, CON149[10], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_TRIG_NOMATCH_M

1_IOACK_UNKNOWN_TYPE_M R/W, CON149[11], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_IOACK_UNKNOWN_TYPE_M

1_IOACK_NOMATCH_M R/W, CON149[12], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_IOACK_NOMATCH_M

1_IOACK_UNEXPECTED_INT_M R/W, CON149[13], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_IOACK_UNEXPECTED_INT_M

1_IOACK_NOMATCH2_M R/W, CON149[14], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_IOACK_NOMATCH2_M

1_STRM_PKT_DROP_M R/W, CON149[15], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_STRM_PKT_DROP_M

1_STRM_NOT_ENOUGH_DAT_M R/W, CON149[16], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_STRM_NOT_ENOUGH_DAT_M

1_STRM_TOO_MUCH_DAT_M R/W, CON149[17], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_STRM_TOO_MUCH_DAT_M

1_STRM_BAD_CRC_M R/W, CON149[18], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_STRM_BAD_CRC_M

1_STRM_OVERFLOW_M R/W, CON149[19], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_STRM_OVERFLOW_M

10.32 CON150

Bit	Name
0	1_UNDER_CURRENT_WP
1	1_OVER_CURRENT_WP
2	1_TRIG_ACK_RCVD_WP
3	1_GPIO_ACK_RCVD_WP
4	1_CTL_ACK_RCVD_WP
5	1_GPIO_RCVD_WP
6	1_TRIG_RCVD_WP
7	1_CTL_RSP_FIFO_OVF_WP
8	1_CTL_REQ_FIFO_OVF_WP
9	1_DOWN_TRIG_RCVD_WP
10	1_TRIG_NOMATCH_WP
11	1_IOACK_UNKNOWN_TYPE_WP
12	1_IOACK_NOMATCH_WP
13	1_IOACK_UNEXPECTED_INT_WP
14	1_IOACK_NOMATCH2_WP
15	1_STRM_PKT_DROP_WP
16	1_STRM_NOT_ENOUGH_DAT_WP
17	1_STRM_TOO_WPUCH_DAT_WP
18	1_STRM_BAD_CRC_WP
19	1_STRM_OVERFLOW_WP
20	1_STRM_CORNER_WP
21	1_SERDES_LOST_ALIGN_WP
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

1_UNDER_CURRENT_WP R/W, CON150[0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_UNDER_CURRENT_WP

1_OVER_CURRENT_WP R/W, CON150[1], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_OVER_CURRENT_WP

1_TRIG_ACK_RCVD_WP R/W, CON150[2], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_TRIG_ACK_RCVD_WP

1_GPIO_ACK_RCVD_WP R/W, CON150[3], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_GPIO_ACK_RCVD_WP

1_CTL_ACK_RCVD_WP R/W, CON150[4], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_CTL_ACK_RCVD_WP

1_GPIO_RCVD_WP R/W, CON150[5], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_GPIO_RCVD_WP

1_TRIG_RCVD_WP R/W, CON150[6], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_TRIG_RCVD_WP

1_CTL_RSP_FIFO_OVF_WP R/W, CON150[7], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_CTL_RSP_FIFO_OVF_WP

1_CTL_REQ_FIFO_OVF_WP R/W, CON150[8], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_CTL_REQ_FIFO_OVF_WP

1_DOWN_TRIG_RCVD_WP R/W, CON150[9], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_DOWN_TRIG_RCVD_WP

1_TRIG_NOMATCH_WP R/W, CON150[10], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_TRIG_NOMATCH_WP

1_IOACK_UNKNOWN_TYPE_WP R/W, CON150[11], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_IOACK_UNKNOWN_TYPE_WP

1_IOACK_NOMATCH_WP R/W, CON150[12], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_IOACK_NOMATCH_WP

1_IOACK_UNEXPECTED_INT_WP R/W, CON150[13], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_IOACK_UNEXPECTED_INT_WP

1_IOACK_NOMATCH2_WP R/W, CON150[14], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_IOACK_NOMATCH2_WP

1_STRM_PKT_DROP_WP R/W, CON150[15], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_STRM_PKT_DROP_WP

1_STRM_NOT_ENOUGH_DAT_WP R/W, CON150[16], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_STRM_NOT_ENOUGH_DAT_WP

1_STRM_TOO_WPUCH_DAT_WP R/W, CON150[17], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_STRM_TOO_WPUCH_DAT_WP

1_STRM_BAD_CRC_WP R/W, CON150[18], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_STRM_BAD_CRC_WP

1_STRM_OVERFLOW_WP R/W, CON150[19], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_STRM_OVERFLOW_WP

10.33 CON152

Bit	Name
0	1_PKT_RCVD_CNT
1	1_PKT_RCVD_CNT
2	1_PKT_RCVD_CNT
3	1_PKT_RCVD_CNT
4	1_PKT_RCVD_CNT
5	1_PKT_RCVD_CNT
6	1_PKT_RCVD_CNT
7	1_PKT_RCVD_CNT
8	1_PKT_RCVD_CNT
9	1_PKT_RCVD_CNT
10	1_PKT_RCVD_CNT
11	1_PKT_RCVD_CNT
12	1_PKT_RCVD_CNT
13	1_PKT_RCVD_CNT
14	1_PKT_RCVD_CNT
15	1_PKT_RCVD_CNT
16	1_PKT_GNT_CNT
17	1_PKT_GNT_CNT
18	1_PKT_GNT_CNT
19	1_PKT_GNT_CNT
20	1_PKT_GNT_CNT
21	1_PKT_GNT_CNT
22	1_PKT_GNT_CNT
23	1_PKT_GNT_CNT
24	1_PKT_GNT_CNT
25	1_PKT_GNT_CNT
26	1_PKT_GNT_CNT
27	1_PKT_GNT_CNT
28	1_PKT_GNT_CNT
29	1_PKT_GNT_CNT
30	1_PKT_GNT_CNT
31	1_PKT_GNT_CNT

1_PKT_RCVD_CNT RO, CON152[15..0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_PKT_RCVD_CNT

1_PKT_GNT_CNT RO, CON152[31..16], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_PKT_GNT_CNT

10.34 CON153

Bit	Name
0	1_PKT_DROP_CNT
1	1_PKT_DROP_CNT
2	1_PKT_DROP_CNT
3	1_PKT_DROP_CNT
4	1_PKT_DROP_CNT
5	1_PKT_DROP_CNT
6	1_PKT_DROP_CNT
7	1_PKT_DROP_CNT
8	1_PKT_DROP_CNT
9	1_PKT_DROP_CNT
10	1_PKT_DROP_CNT
11	1_PKT_DROP_CNT
12	1_PKT_DROP_CNT
13	1_PKT_DROP_CNT
14	1_PKT_DROP_CNT
15	1_PKT_DROP_CNT
16	1_CRC_ERR_CNT
17	1_CRC_ERR_CNT
18	1_CRC_ERR_CNT
19	1_CRC_ERR_CNT
20	1_CRC_ERR_CNT
21	1_CRC_ERR_CNT
22	1_CRC_ERR_CNT
23	1_CRC_ERR_CNT
24	1_CRC_ERR_CNT
25	1_CRC_ERR_CNT
26	1_CRC_ERR_CNT
27	1_CRC_ERR_CNT
28	1_CRC_ERR_CNT
29	1_CRC_ERR_CNT
30	1_CRC_ERR_CNT
31	1_CRC_ERR_CNT

1_PKT_DROP_CNT RO, CON153[15..0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_PKT_DROP_CNT

1_CRC_ERR_CNT RO, CON153[31..16], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_CRC_ERR_CNT

10.35 CON154

Bit	Name
0	1_CXP_TRIG_FALL_SENT
1	1_CXP_TRIG_FALL_SENT
2	1_CXP_TRIG_FALL_SENT
3	1_CXP_TRIG_FALL_SENT
4	1_CXP_TRIG_FALL_SENT
5	1_CXP_TRIG_FALL_SENT
6	1_CXP_TRIG_FALL_SENT
7	1_CXP_TRIG_FALL_SENT
8	1_CXP_TRIG_RISE_SENT
9	1_CXP_TRIG_RISE_SENT
10	1_CXP_TRIG_RISE_SENT
11	1_CXP_TRIG_RISE_SENT
12	1_CXP_TRIG_RISE_SENT
13	1_CXP_TRIG_RISE_SENT
14	1_CXP_TRIG_RISE_SENT
15	1_CXP_TRIG_RISE_SENT
16	1_CXP_TRIG_ACK_CNT
17	1_CXP_TRIG_ACK_CNT
18	1_CXP_TRIG_ACK_CNT
19	1_CXP_TRIG_ACK_CNT
20	1_CXP_TRIG_ACK_CNT
21	1_CXP_TRIG_ACK_CNT
22	1_CXP_TRIG_ACK_CNT
23	1_CXP_TRIG_ACK_CNT
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	1_CXP_TRIG_STATE

1_CXP_TRIG_FALL_SENT RO, CON154[7..0], Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of CXP trigger falling edge trigger packets sent to the camera.

1_CXP_TRIG_RISE_SENT RO, CON154[15..8], Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of CXP trigger rising edge trigger packets sent to the camera.

1_CXP_TRIG_ACK_CNT RO, CON154[23..16], Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of CXP trigger acknowledgments received from the camera.

1_CXP_TRIG_STATE RO, CON154[31], Claxon-CXP, Claxon-FXP, Cyton-CXP

Current state of the uplink CXP trigger signal.

10.36 CON155

Bit	Name
0	1_DT_FALL_RCVD_COUNT
1	1_DT_FALL_RCVD_COUNT
2	1_DT_FALL_RCVD_COUNT
3	1_DT_FALL_RCVD_COUNT
4	1_DT_FALL_RCVD_COUNT
5	1_DT_FALL_RCVD_COUNT
6	1_DT_FALL_RCVD_COUNT
7	1_DT_FALL_RCVD_COUNT
8	1_DT_RISE_RCVD_COUNT
9	1_DT_RISE_RCVD_COUNT
10	1_DT_RISE_RCVD_COUNT
11	1_DT_RISE_RCVD_COUNT
12	1_DT_RISE_RCVD_COUNT
13	1_DT_RISE_RCVD_COUNT
14	1_DT_RISE_RCVD_COUNT
15	1_DT_RISE_RCVD_COUNT
16	1_DT_TRIG_DELAY
17	1_DT_TRIG_DELAY
18	1_DT_TRIG_DELAY
19	1_DT_TRIG_DELAY
20	1_DT_TRIG_DELAY
21	1_DT_TRIG_DELAY
22	1_DT_TRIG_DELAY
23	1_DT_TRIG_DELAY
24	1_DT_FALL_DISABLE
25	1_DT_RISE_DISABLE
26	1_DT_TRIG_NUM
27	1_DT_TRIG_NUM
28	1_DT_TRIG_NUM
29	1_DT_TRIG_NUM
30	1_DT_CLR_CNT
31	1_DT_STATE

1_DT_FALL_RCVD_COUNT RO, CON155[7..0], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of CXP down stream (camera to frame grabber) trigger falling edges.

1_DT_RISE_RCVD_COUNT RO, CON155[15..8], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of CXP down stream (camera to frame grabber) trigger rising edges.

1_DT_TRIG_DELAY RO, CON155[23..16], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

The payload of the CXP down stream (camera to frame grabber) trigger.

1_DT_FALL_DISABLE R/W, CON155[24], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Setting this bit to 1 will disable the falling edge trigger packet from causing an interrupt. It will still be counted and affect DT_STATE.

1_DT_RISE_DISABLE R/W, CON155[25], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Setting this bit to 1 will disable the rising edge trigger packet from causing an interrupt. It will still be counted and affect DT_STATE.

1_DT_TRIG_NUM RO, CON155[29..26], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

When the link is in CXP 2.0 or later mode, these bits capture the TriggerN field of a trigger.

1_DT_CLR_CNT WO, CON155[30], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Writing this register to 1 will clear that counters in this registers.

1_DT_STATE RO, CON155[31], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Current stated of the downstream trigger.

10.37 CON159

Bit	Name
0	1_SERDES_STATE
1	1_SERDES_STATE
2	1_SERDES_STATE
3	1_SERDES_STATE
4	1_SERDES_STATE
5	1_SERDES_STATE
6	1_SERDES_STATE
7	1_SERDES_STATE
8	1_LINK_SPEED
9	1_LINK_SPEED
10	1_LINK_SPEED
11	1_LINK_SPEED
12	1_LINK_SPEED
13	1_LINK_SPEED
14	1_LINK_SPEED
15	1_LINK_SPEED
16	1_LOST_ALIGN_CNT
17	1_LOST_ALIGN_CNT
18	1_LOST_ALIGN_CNT
19	1_LOST_ALIGN_CNT
20	1_LOST_ALIGN_CNT
21	1_LOST_ALIGN_CNT
22	1_LOST_ALIGN_CNT
23	1_LOST_ALIGN_CNT
24	1_LOST_ALIGN_CNT
25	1_LOST_ALIGN_CNT
26	Reserved
27	Reserved
28	1_RESYNC_TOG_MSTR
29	1_RESYNC_TOG
30	1_SERDES_ALIGNED
31	Reserved

1_SERDES_STATE RO, CON159[7..0], Claxon-CXP, Claxon-FXP, Cyton-CXP, Karbon-CXP

Current state of the SERDES associated with this link.

1_LINK_SPEED RO, CON159[15..8], Claxon-CXP, Claxon-FXP, Cyton-CXP, Karbon-CXP

Current speed of the SERDES associated with this link.

1_LOST_ALIGN_CNT RO, CON159[25..16], Claxon-CXP, Claxon-FXP, Cyton-CXP, Karbon-CXP

Number of times the SERDES associated with this link has lost alignment.

1_RESYNC_TOG_MSTR RO, CON159[28], Claxon-CXP, Claxon-FXP, Cyton-CXP, Karbon-CXP

Diagnostics.

1_RESYNC_TOG RO, CON159[29], Claxon-CXP, Claxon-FXP, Cyton-CXP, Karbon-CXP

Diagnostics.

1_SERDES_ALIGNED RO, CON159[30], Claxon-CXP, Claxon-FXP, Cyton-CXP, Karbon-CXP

If this bit is 1 the SERDES associated with this link is aligned.

10.38 CON160

Bit	Name
0	1_RAW_DATA_MODE
1	1_REMOVE_IDLES
2	1_DISABLE_SUB
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	1_MAX_PKT_RCVD
17	1_MAX_PKT_RCVD
18	1_MAX_PKT_RCVD
19	1_MAX_PKT_RCVD
20	1_MAX_PKT_RCVD
21	1_MAX_PKT_RCVD
22	1_MAX_PKT_RCVD
23	1_MAX_PKT_RCVD
24	1_MAX_PKT_RCVD
25	1_MAX_PKT_RCVD
26	1_MAX_PKT_RCVD
27	1_MAX_PKT_RCVD
28	1_MAX_PKT_RCVD
29	1_MAX_PKT_RCVD
30	1_MAX_PKT_RCVD
31	1_MAX_PKT_RCVD

1_RAW_DATA_MODE R/W, CON160[0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_RAW_DATA_MODE

1_REMOVE_IDLES R/W, CON160[1], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_REMOVE_IDLES

1_DISABLE_SUB R/W, CON160[2], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the automatic pixel substitution facility built into the board's receiver circuit. In some very unusual circumstance that low level CXP hardware and perform a pixel substitution if bad data packet is detected. This substitution will still result in a CRC error for the packet, but it will cause less harm than if the original bad packet was received. Setting this bit to 1 will turn this facility off, which is really only useful for testing.

1_MAX_PKT_RCVD R/W, CON160[16], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

The maximum CXP packet that has been received from the camera. Units are 32-bit words. Clear this register by writing a 0 to it.

10.39 CON163

Bit	Name
0	1_SERDES_ERROR_CODE
1	1_SERDES_ERROR_CODE
2	1_SERDES_ERROR_CODE
3	1_SERDES_ERROR_CODE
4	1_SERDES_ERROR_CODE
5	1_SERDES_ERROR_CODE
6	1_SERDES_ERROR_CODE
7	1_SERDES_ERROR_CODE
8	1_SERDES_ERROR_CODE
9	1_SERDES_ERROR_CODE
10	1_SERDES_ERROR_CODE
11	1_SERDES_ERROR_CODE
12	1_SERDES_ERROR_CODE
13	1_SERDES_ERROR_CODE
14	1_SERDES_ERROR_CODE
15	1_SERDES_ERROR_CODE
16	1_SERDES_ERROR_CODE
17	1_SERDES_ERROR_CODE
18	1_SERDES_ERROR_CODE
19	1_SERDES_ERROR_CODE
20	1_SERDES_ERROR_CODE
21	1_SERDES_ERROR_CODE
22	1_SERDES_ERROR_CODE
23	1_SERDES_ERROR_CODE
24	1_SERDES_ERROR_CODE
25	1_SERDES_ERROR_CODE
26	1_SERDES_ERROR_CODE
27	1_SERDES_ERROR_CODE
28	1_SERDES_ERROR_CODE
29	1_SERDES_ERROR_CODE
30	1_SERDES_ERROR_CODE
31	Reserved

1_SERDES_ERROR_CODE

RO, CON163[30..0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_SERDES_ERROR_CODE

10.40 CON168

Bit	Name
0	2_POCXP_EN_POWER
1	2_POCXP_EN_24V_REG
2	2_POCXP_EN_CAM_SENSE
3	2_POCXP_CAM_IS_POCXP
4	2_POCXP_SHORT_DETECTED
5	2_POCXP_OPEN_DETECTED
6	2_POCXP_OVER_DETECTED
7	2_POCXP_OVER_LATCH
8	2_POCXP_UNDER_DETECTED
9	2_POCXP_UNDER_LATCH
10	2_POCXP_24V_OK
11	Reserved
12	2_POCXP_STATE
13	2_POCXP_OVR_AUTO_RESTART
14	2_POCXP_SENSE_BYPASS
15	2_ENABLE_POCXP_SYSTEM
16	2_POCXP_CURRENT_LATCH
17	2_POCXP_CURRENT_LATCH
18	2_POCXP_CURRENT_LATCH
19	2_POCXP_CURRENT_LATCH
20	2_POCXP_CURRENT_LATCH
21	2_POCXP_CURRENT_LATCH
22	2_POCXP_CURRENT_LATCH
23	2_POCXP_CURRENT_LATCH
24	2_POCXP_CURRENT
25	2_POCXP_CURRENT
26	2_POCXP_CURRENT
27	2_POCXP_CURRENT
28	2_POCXP_CURRENT
29	2_POCXP_CURRENT
30	2_POCXP_CURRENT
31	2_POCXP_CURRENT

2_POCXP_EN_POWER R/W, CON168[0], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_EN_POWER

2_POCXP_EN_24V_REG R/W, CON168[1], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_EN_24V_REG

2_POCXP_EN_CAM_SENSE RO, CON168[2], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_EN_CAM_SENSE

2_POCXP_CAM_IS_POCXP RO, CON168[3], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_CAM_IS_POCXP

2_POCXP_SHORT_DETECTED RO, CON168[4], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_SHORT_DETECTED

2_POCXP_OPEN_DETECTED RO, CON168[5], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_OPEN_DETECTED

2_POCXP_OVER_DETECTED RO, CON168[6], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_OVER_DETECTED

2_POCXP_OVER_LATCH RO, CON168[7], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_OVER_LATCH

2_POCXP_UNDER_DETECTED RO, CON168[8], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_UNDER_DETECTED

2_POCXP_UNDER_LATCH RO, CON168[9], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_UNDER_LATCH

2_POCXP_24V_OK RO, CON168[10], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_24V_OK

2_POCXP_STATE RO, CON168[12], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_STATE.

2_POCXP_OVR_AUTO_RESTART RW, CON168[13], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_OVR_AUTO_RESTART.

2_POCXP_SENSE_BYPASS RW, CON168[14], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_SENSE_BYPASS.

2_ENABLE_POCXP_SYSTEM RW, CON168[15], Claxon-CXP, Cyton-CXP

See description of 0_ENABLE_POCXP_SYSTEM.

2_POCXP_CURRENT_LATCH RO, CON168[23..16], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_CURRENT_LATCH

2_POCXP_CURRENT RO, CON168[31..24], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_CURRENT

10.41 CON169

Bit	Name
0	2_POCXP_OVER_TIMER
1	2_POCXP_OVER_TIMER
2	2_POCXP_OVER_TIMER
3	2_POCXP_OVER_TIMER
4	2_POCXP_OVER_TIMER
5	2_POCXP_OVER_TIMER
6	2_POCXP_OVER_TIMER
7	2_POCXP_OVER_TIMER
8	2_POCXP_OVER_TIMER
9	2_POCXP_OVER_TIMER
10	2_POCXP_OVER_TIMER
11	2_POCXP_OVER_TIMER
12	2_POCXP_OVER_TIMER
13	2_POCXP_OVER_TIMER
14	2_POCXP_OVER_TIMER
15	2_POCXP_OVER_TIMER
16	2_POCXP_OVER_TIMER
17	2_POCXP_OVER_TIMER
18	2_POCXP_OVER_TIMER
19	2_POCXP_OVER_TIMER
20	2_POCXP_OVER_TIMER
21	2_POCXP_OVER_TIMER
22	2_POCXP_OVER_TIMER
23	2_POCXP_OVER_TIMER
24	2_POCXP_OVER_TIMER
25	2_POCXP_OVER_TIMER
26	2_POCXP_OVER_TIMER
27	2_POCXP_OVER_TIMER
28	2_POCXP_OVER_TIMER
29	2_POCXP_OVER_TIMER
30	2_POCXP_OVER_TIMER
31	2_POCXP_OVER_TIMER

2_POCXP_OVER_TIMER R/W, CON169[31..0], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_OVER_TIMER

10.42 CON170

Bit	Name
0	2_POCXP_UNDER_TIMER
1	2_POCXP_UNDER_TIMER
2	2_POCXP_UNDER_TIMER
3	2_POCXP_UNDER_TIMER
4	2_POCXP_UNDER_TIMER
5	2_POCXP_UNDER_TIMER
6	2_POCXP_UNDER_TIMER
7	2_POCXP_UNDER_TIMER
8	2_POCXP_UNDER_TIMER
9	2_POCXP_UNDER_TIMER
10	2_POCXP_UNDER_TIMER
11	2_POCXP_UNDER_TIMER
12	2_POCXP_UNDER_TIMER
13	2_POCXP_UNDER_TIMER
14	2_POCXP_UNDER_TIMER
15	2_POCXP_UNDER_TIMER
16	2_POCXP_UNDER_TIMER
17	2_POCXP_UNDER_TIMER
18	2_POCXP_UNDER_TIMER
19	2_POCXP_UNDER_TIMER
20	2_POCXP_UNDER_TIMER
21	2_POCXP_UNDER_TIMER
22	2_POCXP_UNDER_TIMER
23	2_POCXP_UNDER_TIMER
24	2_POCXP_UNDER_TIMER
25	2_POCXP_UNDER_TIMER
26	2_POCXP_UNDER_TIMER
27	2_POCXP_UNDER_TIMER
28	2_POCXP_UNDER_TIMER
29	2_POCXP_UNDER_TIMER
30	2_POCXP_UNDER_TIMER
31	2_POCXP_UNDER_TIMER

2_POCXP_ UNDER_TIMER R/W, CON170[31..0], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_UNDER_TIMER

10.43 CON171

Bit	Name
0	2_COM_RCV_FIFO_SIZE
1	2_COM_RCV_FIFO_SIZE
2	2_COM_RCV_FIFO_SIZE
3	2_COM_RCV_FIFO_SIZE
4	2_COM_RCV_FIFO_SIZE
5	2_COM_RCV_FIFO_SIZE
6	2_COM_RCV_FIFO_SIZE
7	2_COM_RCV_FIFO_SIZE
8	2_COM_RCV_FIFO_SIZE
9	2_COM_RCV_FIFO_SIZE
10	2_COM_RCV_FIFO_SIZE
11	2_COM_RCV_FIFO_SIZE
12	2_COM_RCV_FIFO_SIZE
13	2_COM_RCV_FIFO_SIZE
14	2_COM_RCV_FIFO_SIZE
15	2_COM_RCV_FIFO_SIZE
16	2_COM_SEND_FIFO_SIZE
17	2_COM_SEND_FIFO_SIZE
18	2_COM_SEND_FIFO_SIZE
19	2_COM_SEND_FIFO_SIZE
20	2_COM_SEND_FIFO_SIZE
21	2_COM_SEND_FIFO_SIZE
22	2_COM_SEND_FIFO_SIZE
23	2_COM_SEND_FIFO_SIZE
24	2_COM_SEND_FIFO_SIZE
25	2_COM_SEND_FIFO_SIZE
26	2_COM_SEND_FIFO_SIZE
27	2_COM_SEND_FIFO_SIZE
28	2_COM_SEND_FIFO_SIZE
29	2_COM_SEND_FIFO_SIZE
30	2_COM_SEND_FIFO_SIZE
31	2_COM_SEND_FIFO_SIZE

2_COM_RCV_FIFO_SIZE RO, CON171[15..0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_RCV_FIFO_SIZE

2_COM_SEND_FIFO_SIZE RO, CON171[31..16], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_SEND_FIFO_SIZE

10.44 CON172

Bit	Name
0	2_COM_SEND_FIFO_CLR
1	2_COM_SEND_GO
2	Reserved
3	2_COM_CLR_SENT_CNT
4	2_COM_SENT_CNT
5	2_COM_SENT_CNT
6	2_COM_SENT_CNT
7	2_COM_SENT_CNT
8	2_COM_SENT_CNT
9	2_COM_SENT_CNT
10	2_COM_SENT_CNT
11	2_COM_SENT_CNT
12	2_COM_UP_SPD_MANUAL
13	2_COM_UP_SPD
14	Reserved
15	Reserved
16	2_COM_SEND_FIFO_CNT
17	2_COM_SEND_FIFO_CNT
18	2_COM_SEND_FIFO_CNT
19	2_COM_SEND_FIFO_CNT
20	2_COM_SEND_FIFO_CNT
21	2_COM_SEND_FIFO_CNT
22	2_COM_SEND_FIFO_CNT
23	2_COM_SEND_FIFO_CNT
24	2_COM_SEND_FIFO_CNT
25	2_COM_SEND_FIFO_CNT
26	2_COM_SEND_FIFO_CNT
27	2_COM_SEND_FIFO_CNT
28	2_COM_SEND_FIFO_CNT
29	2_COM_SEND_FIFO_CNT
30	2_COM_SEND_FIFO_CNT
31	2_COM_SEND_FIFO_CNT

2_COM_SEND_FIFO_CLR WO, CON172[0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_SEND_FIFO_CLR

2_COM_SEND_GO WO, CON172[1], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_SEND_GO

2_COM_CLR_SENT_CNT WO, CON172[3], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_CLR_SENT_CNT

2_COM_SENT_CNT RO, CON172[11..4], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_SENT_CNT.

2_COM_UP_SPD_MANUAL R/W, CON172[12], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_UP_SPD_MANUAL.

2_COM_UP_SPD R/W, CON172[13], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_UP_SPD.

2_COM_SEND_FIFO_CNT RO, CON172[31..16], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_SEND_FIFO_CNT

10.45 CON173

Bit	Name
0	2_COM_SEND_DATA
1	2_COM_SEND_DATA
2	2_COM_SEND_DATA
3	2_COM_SEND_DATA
4	2_COM_SEND_DATA
5	2_COM_SEND_DATA
6	2_COM_SEND_DATA
7	2_COM_SEND_DATA
8	2_COM_SEND_DATA
9	2_COM_SEND_DATA
10	2_COM_SEND_DATA
11	2_COM_SEND_DATA
12	2_COM_SEND_DATA
13	2_COM_SEND_DATA
14	2_COM_SEND_DATA
15	2_COM_SEND_DATA
16	2_COM_SEND_DATA
17	2_COM_SEND_DATA
18	2_COM_SEND_DATA
19	2_COM_SEND_DATA
20	2_COM_SEND_DATA
21	2_COM_SEND_DATA
22	2_COM_SEND_DATA
23	2_COM_SEND_DATA
24	2_COM_SEND_DATA
25	2_COM_SEND_DATA
26	2_COM_SEND_DATA
27	2_COM_SEND_DATA
28	2_COM_SEND_DATA
29	2_COM_SEND_DATA
30	2_COM_SEND_DATA
31	2_COM_SEND_DATA

2_COM_SEND_DATA R/W, CON173[31..0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_SEND_DATA

10.46 CON174

Bit	Name
0	2_COM_RCV_FIFO_CLR
1	Reserved
2	Reserved
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	2_COM_RCV_FIFO_CNT
17	2_COM_RCV_FIFO_CNT
18	2_COM_RCV_FIFO_CNT
19	2_COM_RCV_FIFO_CNT
20	2_COM_RCV_FIFO_CNT
21	2_COM_RCV_FIFO_CNT
22	2_COM_RCV_FIFO_CNT
23	2_COM_RCV_FIFO_CNT
24	2_COM_RCV_FIFO_CNT
25	2_COM_RCV_FIFO_CNT
26	2_COM_RCV_FIFO_CNT
27	2_COM_RCV_FIFO_CNT
28	2_COM_RCV_FIFO_CNT
29	2_COM_RCV_FIFO_CNT
30	2_COM_RCV_FIFO_CNT
31	2_COM_RCV_FIFO_CNT

2_COM_RCV_FIFO_CLR WO, CON174[0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_RCV_FIFO_CLR

2_COM_RCV_FIFO_CNT RO, CON174[31..16], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_RCV_FIFO_CNT

10.47 CON175

Bit	Name
0	2_COM_RCV_DATA
1	2_COM_RCV_DATA
2	2_COM_RCV_DATA
3	2_COM_RCV_DATA
4	2_COM_RCV_DATA
5	2_COM_RCV_DATA
6	2_COM_RCV_DATA
7	2_COM_RCV_DATA
8	2_COM_RCV_DATA
9	2_COM_RCV_DATA
10	2_COM_RCV_DATA
11	2_COM_RCV_DATA
12	2_COM_RCV_DATA
13	2_COM_RCV_DATA
14	2_COM_RCV_DATA
15	2_COM_RCV_DATA
16	2_COM_RCV_DATA
17	2_COM_RCV_DATA
18	2_COM_RCV_DATA
19	2_COM_RCV_DATA
20	2_COM_RCV_DATA
21	2_COM_RCV_DATA
22	2_COM_RCV_DATA
23	2_COM_RCV_DATA
24	2_COM_RCV_DATA
25	2_COM_RCV_DATA
26	2_COM_RCV_DATA
27	2_COM_RCV_DATA
28	2_COM_RCV_DATA
29	2_COM_RCV_DATA
30	2_COM_RCV_DATA
31	2_COM_RCV_DATA

2_COM_RCV_DATA RO, CON175[31..0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_RCV_DATA

10.48 CON179

Bit	Name
0	2_LINK_INT_DEST
1	2_LINK_INT_DEST
2	Reserved
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

2_LINK_INT_DEST R/W, CON179[1..0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_LINK_INT_DEST

10.49 CON180

Bit	Name
0	2_UNDER_CURRENT
1	2_OVER_CURRENT
2	2_TRIG_ACK_RCVD
3	2_CTL_ACK_VER_ERR
4	2_CTL_ACK_RCVD
5	2_EVENT_RCVD
6	2_EVENT_FIFO_OVERFLOW
7	2_CTL_RSP_FIFO_OVF
8	2_CTL_REQ_FIFO_OVF
9	2_DOWN_TRIG_RCVD
10	2_TRIG_NOMATCH
11	2_IOACK_UNKNOWN_TYPE
12	2_IOACK_NOMATCH
13	2_HB_ERROR
14	2_HB_RCVD
15	2_STRM_PKT_DROP
16	2_STRM_NOT_ENOUGH_DAT
17	2_STRM_TOO_MUCH_DAT
18	2_STRM_BAD_CRC
19	2_STRM_OVERFLOW
20	2_STRM_CORNER
21	2_SERDES_LOST_ALIGN
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

2_UNDER_CURRENT R/W, CON180[0], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Undercurrent detected by POCXP controller interrupt.

2_OVER_CURRENT R/W, CON180[1], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Overcurrent detected by POCXP controller interrupt.

2_TRIG_ACK_RCVD R/W, CON180[2], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Trigger acknowledgment received from device interrupt.

2_CTL_ACK_VER_ERR R/W, CON180[3], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

There is a version mismatch between an incoming control packet acknowledgment and the board's current version settings. For example, the board could be in CXP 2.0 mode and it receives a CXP 1.1 acknowledgment, then this interrupt would be set.

2_CTL_ACK_RCVD R/W, CON180[4], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Control packet acknowledgment received from device interrupt.

2_EVENT_RCVD R/W, CON180[5], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

CXP 2.0 Event has been received.

2_EVENT_FIFO_OVERFLOW R/W, CON180[6], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

The CXP 2.0 Event FIFO has overflowed.

2_CTL_RSP_FIFO_OVF R/W, CON180[7], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Overflow detected in the control response FIFO (Device to host direction) interrupt.

2_CTL_REQ_FIFO_OVF R/W, CON180[8], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Overflow detected in control request FIFO (Host to device direction) interrupt.

2_DOWN_TRIG_RCVD R/W, CON180[9], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Received CXP trigger from camera.

2_TRIG_NOMATCH R/W, CON180[10], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Problem decoding a trigger packet from device interrupt.

2_IOACK_UNKNOWN_TYPE R/W, CON180[11], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Problem decoding an IO acknowledgment from device interrupt.

2_IOACK_NOMATCH R/W, CON180[12], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Problem decoding an IO acknowledgment from device interrupt.

2_HB_ERROR R/W, CON180[13], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

There is an error decoding a CXP 2.0 Heart Beat Packet.

2_HB_RCVD R/W, CON180[14], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

A CXP 2.0 Heart Beat packet has been received.

2_STRM_PKT_DROP R/W, CON180[15], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Problem decoding a stream packet header, remainder of packet is dropped interrupt.

2_STRM_NOT_ENOUGH_DAT R/W, CON180[16], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Not implemented, reserved interrupt.

2_STRM_TOO_MUCH_DAT R/W, CON180[17], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Not implemented, reserved interrupt.

2_STRM_BAD_CRC R/W, CON180[18], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

CRC error detected in stream packet interrupt.

2_STRM_OVERFLOW R/W, CON180[19], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Overflow in the stream packet buffer interrupt.

2_STRM_CORNER R/W, CON180[19], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Not implemented, reserved interrupt.

2_SERDES_LOST_ALIGN R/W, CON180[21], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

SERDES lost alignment interrupt.

10.50 CON181

Bit	Name
0	2_UNDER_CURRENT_M
1	2_OVER_CURRENT_M
2	2_TRIG_ACK_RCVD_M
3	2_GPIO_ACK_RCVD_M
4	2_CTL_ACK_RCVD_M
5	2_GPIO_RCVD_M
6	2_TRIG_RCVD_M
7	2_CTL_RSP_FIFO_OVF_M
8	2_CTL_REQ_FIFO_OVF_M
9	2_DOWN_TRIG_RCVD_M
10	2_TRIG_NOMATCH_M
11	2_IOACK_UNKNOWN_TYPE_M
12	2_IOACK_NOMATCH_M
13	2_IOACK_UNEXPECTED_INT_M
14	2_IOACK_NOMATCH2_M
15	2_STRM_PKT_DROP_M
16	2_STRM_NOT_ENOUGH_DAT_M
17	2_STRM_TOO_MUCH_DAT_M
18	2_STRM_BAD_CRC_M
19	2_STRM_OVERFLOW_M
20	2_STRM_CORNER_M
21	2_SERDES_LOST_ALIGN_M
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

2_UNDER_CURRENT_M R/W, CON181[0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_UNDER_CURRENT_M

2_OVER_CURRENT_M R/W, CON181[1], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_OVER_CURRENT_M

2_TRIG_ACK_RCVD_M R/W, CON181[2], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_TRIG_ACK_RCVD_M

2_GPIO_ACK_RCVD_M R/W, CON181[3], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_GPIO_ACK_RCVD_M

2_CTL_ACK_RCVD_M R/W, CON181[4], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_CTL_ACK_RCVD_M

2_GPIO_RCVD_M R/W, CON181[5], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_GPIO_RCVD_M

2_TRIG_RCVD_M R/W, CON181[6], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_TRIG_RCVD_M

2_CTL_RSP_FIFO_OVF_M R/W, CON181[7], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_CTL_RSP_FIFO_OVF_M

2_CTL_REQ_FIFO_OVF_M R/W, CON181[8], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_CTL_REQ_FIFO_OVF_M

2_DOWN_TRIG_RCVD_M R/W, CON181[9], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_DOWN_TRIG_RCVD_M

2_TRIG_NOMATCH_M R/W, CON181[10], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_TRIG_NOMATCH_M

2_IOACK_UNKNOWN_TYPE_M R/W, CON181[11], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_IOACK_UNKNOWN_TYPE_M

2_IOACK_NOMATCH_M R/W, CON181[12], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_IOACK_NOMATCH_M

2_IOACK_UNEXPECTED_INT_M R/W, CON181[13], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_IOACK_UNEXPECTED_INT_M

2_IOACK_NOMATCH2_M R/W, CON181[14], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_IOACK_NOMATCH2_M

2_STRM_PKT_DROP_M R/W, CON181[15], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_STRM_PKT_DROP_M

2_STRM_NOT_ENOUGH_DAT_M R/W, CON181[16], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_STRM_NOT_ENOUGH_DAT_M

2_STRM_TOO_MUCH_DAT_M R/W, CON181[17], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_STRM_TOO_MUCH_DAT_M

2_STRM_BAD_CRC_M R/W, CON181[18], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_STRM_BAD_CRC_M

2_STRM_OVERFLOW_M R/W, CON181[19], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_STRM_OVERFLOW_M

10.51 CON182

Bit	Name
0	2_UNDER_CURRENT_WP
1	2_OVER_CURRENT_WP
2	2_TRIG_ACK_RCVD_WP
3	2_GPIO_ACK_RCVD_WP
4	2_CTL_ACK_RCVD_WP
5	2_GPIO_RCVD_WP
6	2_TRIG_RCVD_WP
7	2_CTL_RSP_FIFO_OVF_WP
8	2_CTL_REQ_FIFO_OVF_WP
9	2_DOWN_TRIG_RCVD_WP
10	2_TRIG_NOMATCH_WP
11	2_IOACK_UNKNOWN_TYPE_WP
12	2_IOACK_NOMATCH_WP
13	2_IOACK_UNEXPECTED_INT_WP
14	2_IOACK_NOMATCH2_WP
15	2_STRM_PKT_DROP_WP
16	2_STRM_NOT_ENOUGH_DAT_WP
17	2_STRM_TOO_WPUCH_DAT_WP
18	2_STRM_BAD_CRC_WP
19	2_STRM_OVERFLOW_WP
20	2_STRM_CORNER_WP
21	2_SERDES_LOST_ALIGN_WP
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

2_UNDER_CURRENT_WP R/W, CON182[0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_UNDER_CURRENT_WP

2_OVER_CURRENT_WP R/W, CON182[1], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_OVER_CURRENT_WP

2_TRIG_ACK_RCVD_WP R/W, CON182[2], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_TRIG_ACK_RCVD_WP

2_GPIO_ACK_RCVD_WP R/W, CON182[3], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_GPIO_ACK_RCVD_WP

2_CTL_ACK_RCVD_WP R/W, CON182[4], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_CTL_ACK_RCVD_WP

2_GPIO_RCVD_WP R/W, CON182[5], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_GPIO_RCVD_WP

2_TRIG_RCVD_WP R/W, CON182[6], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_TRIG_RCVD_WP

2_CTL_RSP_FIFO_OVF_WP R/W, CON182[7], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_CTL_RSP_FIFO_OVF_WP

2_CTL_REQ_FIFO_OVF_WP R/W, CON182[8], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_CTL_REQ_FIFO_OVF_WP

2_DOWN_TRIG_RCVD_WP R/W, CON182[9], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_DOWN_TRIG_RCVD_WP

2_TRIG_NOMATCH_WP R/W, CON182[10], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_TRIG_NOMATCH_WP

2_IOACK_UNKNOWN_TYPE_WP R/W, CON182[11], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_IOACK_UNKNOWN_TYPE_WP

2_IOACK_NOMATCH_WP R/W, CON182[12], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_IOACK_NOMATCH_WP

2_IOACK_UNEXPECTED_INT_WP R/W, CON182[13], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_IOACK_UNEXPECTED_INT_WP

2_IOACK_NOMATCH2_WP R/W, CON182[14], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_IOACK_NOMATCH2_WP

2_STRM_PKT_DROP_WP R/W, CON182[15], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_STRM_PKT_DROP_WP

2_STRM_NOT_ENOUGH_DAT_WP R/W, CON182[16], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_STRM_NOT_ENOUGH_DAT_WP

2_STRM_TOO_WPUCH_DAT_WP R/W, CON182[17], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_STRM_TOO_WPUCH_DAT_WP

2_STRM_BAD_CRC_WP R/W, CON182[18], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_STRM_BAD_CRC_WP

2_STRM_OVERFLOW_WP R/W, CON182[19], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_STRM_OVERFLOW_WP

10.52 CON184

Bit	Name
0	2_PKT_RCVD_CNT
1	2_PKT_RCVD_CNT
2	2_PKT_RCVD_CNT
3	2_PKT_RCVD_CNT
4	2_PKT_RCVD_CNT
5	2_PKT_RCVD_CNT
6	2_PKT_RCVD_CNT
7	2_PKT_RCVD_CNT
8	2_PKT_RCVD_CNT
9	2_PKT_RCVD_CNT
10	2_PKT_RCVD_CNT
11	2_PKT_RCVD_CNT
12	2_PKT_RCVD_CNT
13	2_PKT_RCVD_CNT
14	2_PKT_RCVD_CNT
15	2_PKT_RCVD_CNT
16	2_PKT_GNT_CNT
17	2_PKT_GNT_CNT
18	2_PKT_GNT_CNT
19	2_PKT_GNT_CNT
20	2_PKT_GNT_CNT
21	2_PKT_GNT_CNT
22	2_PKT_GNT_CNT
23	2_PKT_GNT_CNT
24	2_PKT_GNT_CNT
25	2_PKT_GNT_CNT
26	2_PKT_GNT_CNT
27	2_PKT_GNT_CNT
28	2_PKT_GNT_CNT
29	2_PKT_GNT_CNT
30	2_PKT_GNT_CNT
31	2_PKT_GNT_CNT

2_PKT_RCVD_CNT RO, CON184[15..0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_PKT_RCVD_CNT

2_PKT_GNT_CNT RO, CON184[31..16], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_PKT_GNT_CNT

10.53 CON185

Bit	Name
0	2_PKT_DROP_CNT
1	2_PKT_DROP_CNT
2	2_PKT_DROP_CNT
3	2_PKT_DROP_CNT
4	2_PKT_DROP_CNT
5	2_PKT_DROP_CNT
6	2_PKT_DROP_CNT
7	2_PKT_DROP_CNT
8	2_PKT_DROP_CNT
9	2_PKT_DROP_CNT
10	2_PKT_DROP_CNT
11	2_PKT_DROP_CNT
12	2_PKT_DROP_CNT
13	2_PKT_DROP_CNT
14	2_PKT_DROP_CNT
15	2_PKT_DROP_CNT
16	2_CRC_ERR_CNT
17	2_CRC_ERR_CNT
18	2_CRC_ERR_CNT
19	2_CRC_ERR_CNT
20	2_CRC_ERR_CNT
21	2_CRC_ERR_CNT
22	2_CRC_ERR_CNT
23	2_CRC_ERR_CNT
24	2_CRC_ERR_CNT
25	2_CRC_ERR_CNT
26	2_CRC_ERR_CNT
27	2_CRC_ERR_CNT
28	2_CRC_ERR_CNT
29	2_CRC_ERR_CNT
30	2_CRC_ERR_CNT
31	2_CRC_ERR_CNT

2_PKT_DROP_CNT RO, CON185[15..0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_PKT_DROP_CNT

2_CRC_ERR_CNT RO, CON185[31..16], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_CRC_ERR_CNT

10.54 CON186

Bit	Name
0	2_CXP_TRIG_FALL_SENT
1	2_CXP_TRIG_FALL_SENT
2	2_CXP_TRIG_FALL_SENT
3	2_CXP_TRIG_FALL_SENT
4	2_CXP_TRIG_FALL_SENT
5	2_CXP_TRIG_FALL_SENT
6	2_CXP_TRIG_FALL_SENT
7	2_CXP_TRIG_FALL_SENT
8	2_CXP_TRIG_RISE_SENT
9	2_CXP_TRIG_RISE_SENT
10	2_CXP_TRIG_RISE_SENT
11	2_CXP_TRIG_RISE_SENT
12	2_CXP_TRIG_RISE_SENT
13	2_CXP_TRIG_RISE_SENT
14	2_CXP_TRIG_RISE_SENT
15	2_CXP_TRIG_RISE_SENT
16	2_CXP_TRIG_ACK_CNT
17	2_CXP_TRIG_ACK_CNT
18	2_CXP_TRIG_ACK_CNT
19	2_CXP_TRIG_ACK_CNT
20	2_CXP_TRIG_ACK_CNT
21	2_CXP_TRIG_ACK_CNT
22	2_CXP_TRIG_ACK_CNT
23	2_CXP_TRIG_ACK_CNT
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	2_CXP_TRIG_STATE

2_CXP_TRIG_FALL_SENT RO, CON186[7..0], Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of CXP trigger falling edge trigger packets sent to the camera.

2_CXP_TRIG_RISE_SENT RO, CON186[15..8], Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of CXP trigger rising edge trigger packets sent to the camera.

2_CXP_TRIG_ACK_CNT RO, CON186[23..16], Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of CXP trigger acknowledgments received from the camera.

2_CXP_TRIG_STATE RO, CON186[31], Claxon-CXP, Claxon-FXP, Cyton-CXP

Current state of the uplink CXP trigger signal.

10.55 CON187

Bit	Name
0	2_DT_FALL_RCVD_COUNT
1	2_DT_FALL_RCVD_COUNT
2	2_DT_FALL_RCVD_COUNT
3	2_DT_FALL_RCVD_COUNT
4	2_DT_FALL_RCVD_COUNT
5	2_DT_FALL_RCVD_COUNT
6	2_DT_FALL_RCVD_COUNT
7	2_DT_FALL_RCVD_COUNT
8	2_DT_RISE_RCVD_COUNT
9	2_DT_RISE_RCVD_COUNT
10	2_DT_RISE_RCVD_COUNT
11	2_DT_RISE_RCVD_COUNT
12	2_DT_RISE_RCVD_COUNT
13	2_DT_RISE_RCVD_COUNT
14	2_DT_RISE_RCVD_COUNT
15	2_DT_RISE_RCVD_COUNT
16	2_DT_TRIG_DELAY
17	2_DT_TRIG_DELAY
18	2_DT_TRIG_DELAY
19	2_DT_TRIG_DELAY
20	2_DT_TRIG_DELAY
21	2_DT_TRIG_DELAY
22	2_DT_TRIG_DELAY
23	2_DT_TRIG_DELAY
24	2_DT_FALL_DISABLE
25	2_DT_RISE_DISABLE
26	2_DT_TRIG_NUM
27	2_DT_TRIG_NUM
28	2_DT_TRIG_NUM
29	2_DT_TRIG_NUM
30	2_DT_CLR_CNT
31	2_DT_STATE

2_DT_FALL_RCVD_COUNT RO, CON187[7..0], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of CXP down stream (camera to frame grabber) trigger falling edges.

2_DT_RISE_RCVD_COUNT RO, CON187[15..8], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of CXP down stream (camera to frame grabber) trigger rising edges.

2_DT_TRIG_DELAY RO, CON187[23..16], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

The payload of the CXP down stream (camera to frame grabber) trigger.

2_DT_FALL_DISABLE R/W, CON187[24], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Setting this bit to 1 will disable the falling edge trigger packet from causing an interrupt. It will still be counted and affect DT_STATE.

2_DT_RISE_DISABLE R/W, CON187[25], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Setting this bit to 1 will disable the rising edge trigger packet from causing an interrupt. It will still be counted and affect DT_STATE.

2_DT_TRIG_NUM RO, CON187[29..26], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

When the link is in CXP 2.0 or later mode, these bits capture the TriggerN field of a trigger.

2_DT_CLR_CNT WO, CON187[30], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Writing this register to 1 will clear that counters in this registers.

2_DT_STATE RO, CON187[31], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Current stated of the downstream trigger.

10.56 CON191

Bit	Name
0	2_SERDES_STATE
1	2_SERDES_STATE
2	2_SERDES_STATE
3	2_SERDES_STATE
4	2_SERDES_STATE
5	2_SERDES_STATE
6	2_SERDES_STATE
7	2_SERDES_STATE
8	2_LINK_SPEED
9	2_LINK_SPEED
10	2_LINK_SPEED
11	2_LINK_SPEED
12	2_LINK_SPEED
13	2_LINK_SPEED
14	2_LINK_SPEED
15	2_LINK_SPEED
16	2_LOST_ALIGN_CNT
17	2_LOST_ALIGN_CNT
18	2_LOST_ALIGN_CNT
19	2_LOST_ALIGN_CNT
20	2_LOST_ALIGN_CNT
21	2_LOST_ALIGN_CNT
22	2_LOST_ALIGN_CNT
23	2_LOST_ALIGN_CNT
24	2_LOST_ALIGN_CNT
25	2_LOST_ALIGN_CNT
26	Reserved
27	Reserved
28	2_RESYNC_TOG_MSTR
29	2_RESYNC_TOG
30	2_SERDES_ALIGNED
31	Reserved

2_SERDES_STATE RO, CON191[7..0], Claxon-CXP, Claxon-FXP, Cyton-CXP, Karbon-CXP

Current state of the SERDES associated with this link.

2_LINK_SPEED RO, CON191[15..8], Claxon-CXP, Claxon-FXP, Cyton-CXP, Karbon-CXP

Current speed of the SERDES associated with this link.

2_LOST_ALIGN_CNT RO, CON191[25..16], Claxon-CXP, Claxon-FXP, Cyton-CXP, Karbon-CXP

Describe 2_LOST_ALIGN_CNT here.

2_RESYNC_TOG_MSTR RO, CON191[28], Claxon-CXP, Claxon-FXP, Cyton-CXP, Karbon-CXP

Diagnostics.

2_RESYNC_TOG RO, CON191[29], Claxon-CXP, Claxon-FXP, Cyton-CXP, Karbon-CXP

Diagnostics.

2_SERDES_ALIGNED RO, CON191[30], Claxon-CXP, Claxon-FXP, Cyton-CXP, Karbon-CXP

If this bit is 1 the SERDES associated with this link is aligned.

10.57 CON192

Bit	Name
0	2_RAW_DATA_MODE
1	2_REMOVE_IDLES
2	2_DISABLE_SUB
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	1_MAX_PKT_RCVD
17	1_MAX_PKT_RCVD
18	1_MAX_PKT_RCVD
19	1_MAX_PKT_RCVD
20	1_MAX_PKT_RCVD
21	1_MAX_PKT_RCVD
22	1_MAX_PKT_RCVD
23	1_MAX_PKT_RCVD
24	1_MAX_PKT_RCVD
25	1_MAX_PKT_RCVD
26	1_MAX_PKT_RCVD
27	1_MAX_PKT_RCVD
28	1_MAX_PKT_RCVD
29	1_MAX_PKT_RCVD
30	1_MAX_PKT_RCVD
31	1_MAX_PKT_RCVD

2_RAW_DATA_MODE R/W, CON192[0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_RAW_DATA_MODE

2_REMOVE_IDLES R/W, CON192[1], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_REMOVE_IDLES

2_DISABLE_SUB R/W, CON192[2], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the automatic pixel substitution facility built into the board's receiver circuit. In some very unusual circumstance that low level CXP hardware and perform a pixel substitution if bad data packet is detected. This substitution will still result in a CRC error for the packet, but it will cause less harm than if the original bad packet was received. Setting this bit to 1 will turn this facility off, which is really only useful for testing.

2_MAX_PKT_RCVD R/W, CON192[16], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

The maximum CXP packet that has been received from the camera. Units are 32-bit words. Clear this register by writing a 0 to it.

10.58 CON195

Bit	Name
0	2_SERDES_ERROR_CODE
1	2_SERDES_ERROR_CODE
2	2_SERDES_ERROR_CODE
3	2_SERDES_ERROR_CODE
4	2_SERDES_ERROR_CODE
5	2_SERDES_ERROR_CODE
6	2_SERDES_ERROR_CODE
7	2_SERDES_ERROR_CODE
8	2_SERDES_ERROR_CODE
9	2_SERDES_ERROR_CODE
10	2_SERDES_ERROR_CODE
11	2_SERDES_ERROR_CODE
12	2_SERDES_ERROR_CODE
13	2_SERDES_ERROR_CODE
14	2_SERDES_ERROR_CODE
15	2_SERDES_ERROR_CODE
16	2_SERDES_ERROR_CODE
17	2_SERDES_ERROR_CODE
18	2_SERDES_ERROR_CODE
19	2_SERDES_ERROR_CODE
20	2_SERDES_ERROR_CODE
21	2_SERDES_ERROR_CODE
22	2_SERDES_ERROR_CODE
23	2_SERDES_ERROR_CODE
24	2_SERDES_ERROR_CODE
25	2_SERDES_ERROR_CODE
26	2_SERDES_ERROR_CODE
27	2_SERDES_ERROR_CODE
28	2_SERDES_ERROR_CODE
29	2_SERDES_ERROR_CODE
30	2_SERDES_ERROR_CODE
31	Reserved

2_SERDES_ERROR_CODE RO, CON195[30..0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_SERDES_ERROR_CODE

10.59 CON200

Bit	Name
0	3_POCXP_EN_POWER
1	3_POCXP_EN_24V_REG
2	3_POCXP_EN_CAM_SENSE
3	3_POCXP_CAM_IS_POCXP
4	3_POCXP_SHORT_DETECTED
5	3_POCXP_OPEN_DETECTED
6	3_POCXP_OVER_DETECTED
7	3_POCXP_OVER_LATCH
8	3_POCXP_UNDER_DETECTED
9	3_POCXP_UNDER_LATCH
10	3_POCXP_24V_OK
11	Reserved
12	3_POCXP_STATE
13	3_POCXP_OVR_AUTO_RESTART
14	3_POCXP_SENSE_BYPASS
15	3_ENABLE_POCXP_SYSTEM
16	3_POCXP_CURRENT_LATCH
17	3_POCXP_CURRENT_LATCH
18	3_POCXP_CURRENT_LATCH
19	3_POCXP_CURRENT_LATCH
20	3_POCXP_CURRENT_LATCH
21	3_POCXP_CURRENT_LATCH
22	3_POCXP_CURRENT_LATCH
23	3_POCXP_CURRENT_LATCH
24	3_POCXP_CURRENT
25	3_POCXP_CURRENT
26	3_POCXP_CURRENT
27	3_POCXP_CURRENT
28	3_POCXP_CURRENT
29	3_POCXP_CURRENT
30	3_POCXP_CURRENT
31	3_POCXP_CURRENT

3_POCXP_EN_POWER RO, CON200[0], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_EN_POWER

3_POCXP_EN_24V_REG RO, CON200[1], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_EN_24V_REG

3_POCXP_EN_CAM_SENSE RO, CON200[2], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_EN_CAM_SENSE

3_POCXP_CAM_IS_POCXP RO, CON200[3], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_CAM_IS_POCXP

3_POCXP_SHORT_DETECTED RO, CON200[4], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_SHORT_DETECTED

3_POCXP_OPEN_DETECTED RO, CON200[5], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_OPEN_DETECTED

3_POCXP_OVER_DETECTED RO, CON200[6], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_OVER_DETECTED

3_POCXP_OVER_LATCH RO, CON200[7], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_OVER_LATCH

3_POCXP_UNDER_DETECTED RO, CON200[8], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_UNDER_DETECTED

3_POCXP_UNDER_LATCH RO, CON200[9], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_UNDER_LATCH

3_POCXP_24V_OK RO, CON200[10], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_24V_OK

3_POCXP_STATE RO, CON200[12], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_STATE.

3_POCXP_OVR_AUTO_RESTART RW, CON200[13], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_OVR_AUTO_RESTART.

3_POCXP_SENSE_BYPASS RW, CON200[14], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_SENSE_BYPASS.

3_ENABLE_POCXP_SYSTEM RW, CON200[15], Claxon-CXP, Cyton-CXP

See description of 0_ENABLE_POCXP_SYSTEM.

3_POCXP_CURRENT_LATCH RO, CON200[23..16], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_CURRENT_LATCH

3_POCXP_CURRENT RO, CON200[31..24], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_CURRENT

10.60 CON201

Bit	Name
0	3_POCXP_OVER_TIMER
1	3_POCXP_OVER_TIMER
2	3_POCXP_OVER_TIMER
3	3_POCXP_OVER_TIMER
4	3_POCXP_OVER_TIMER
5	3_POCXP_OVER_TIMER
6	3_POCXP_OVER_TIMER
7	3_POCXP_OVER_TIMER
8	3_POCXP_OVER_TIMER
9	3_POCXP_OVER_TIMER
10	3_POCXP_OVER_TIMER
11	3_POCXP_OVER_TIMER
12	3_POCXP_OVER_TIMER
13	3_POCXP_OVER_TIMER
14	3_POCXP_OVER_TIMER
15	3_POCXP_OVER_TIMER
16	3_POCXP_OVER_TIMER
17	3_POCXP_OVER_TIMER
18	3_POCXP_OVER_TIMER
19	3_POCXP_OVER_TIMER
20	3_POCXP_OVER_TIMER
21	3_POCXP_OVER_TIMER
22	3_POCXP_OVER_TIMER
23	3_POCXP_OVER_TIMER
24	3_POCXP_OVER_TIMER
25	3_POCXP_OVER_TIMER
26	3_POCXP_OVER_TIMER
27	3_POCXP_OVER_TIMER
28	3_POCXP_OVER_TIMER
29	3_POCXP_OVER_TIMER
30	3_POCXP_OVER_TIMER
31	3_POCXP_OVER_TIMER

3_POCXP_OVER_TIMER R/W, CON201[31..0], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_OVER_TIMER

10.61 CON202

Bit	Name
0	3_POCXP_UNDER_TIMER
1	3_POCXP_UNDER_TIMER
2	3_POCXP_UNDER_TIMER
3	3_POCXP_UNDER_TIMER
4	3_POCXP_UNDER_TIMER
5	3_POCXP_UNDER_TIMER
6	3_POCXP_UNDER_TIMER
7	3_POCXP_UNDER_TIMER
8	3_POCXP_UNDER_TIMER
9	3_POCXP_UNDER_TIMER
10	3_POCXP_UNDER_TIMER
11	3_POCXP_UNDER_TIMER
12	3_POCXP_UNDER_TIMER
13	3_POCXP_UNDER_TIMER
14	3_POCXP_UNDER_TIMER
15	3_POCXP_UNDER_TIMER
16	3_POCXP_UNDER_TIMER
17	3_POCXP_UNDER_TIMER
18	3_POCXP_UNDER_TIMER
19	3_POCXP_UNDER_TIMER
20	3_POCXP_UNDER_TIMER
21	3_POCXP_UNDER_TIMER
22	3_POCXP_UNDER_TIMER
23	3_POCXP_UNDER_TIMER
24	3_POCXP_UNDER_TIMER
25	3_POCXP_UNDER_TIMER
26	3_POCXP_UNDER_TIMER
27	3_POCXP_UNDER_TIMER
28	3_POCXP_UNDER_TIMER
29	3_POCXP_UNDER_TIMER
30	3_POCXP_UNDER_TIMER
31	3_POCXP_UNDER_TIMER

3_POCXP_UNDER_TIMER R/W, CON202[31..0], Claxon-CXP, Cyton-CXP

See description of 0_POCXP_UNDER_TIMER

10.62 CON203

Bit	Name
0	3_COM_RCV_FIFO_SIZE
1	3_COM_RCV_FIFO_SIZE
2	3_COM_RCV_FIFO_SIZE
3	3_COM_RCV_FIFO_SIZE
4	3_COM_RCV_FIFO_SIZE
5	3_COM_RCV_FIFO_SIZE
6	3_COM_RCV_FIFO_SIZE
7	3_COM_RCV_FIFO_SIZE
8	3_COM_RCV_FIFO_SIZE
9	3_COM_RCV_FIFO_SIZE
10	3_COM_RCV_FIFO_SIZE
11	3_COM_RCV_FIFO_SIZE
12	3_COM_RCV_FIFO_SIZE
13	3_COM_RCV_FIFO_SIZE
14	3_COM_RCV_FIFO_SIZE
15	3_COM_RCV_FIFO_SIZE
16	3_COM_SEND_FIFO_SIZE
17	3_COM_SEND_FIFO_SIZE
18	3_COM_SEND_FIFO_SIZE
19	3_COM_SEND_FIFO_SIZE
20	3_COM_SEND_FIFO_SIZE
21	3_COM_SEND_FIFO_SIZE
22	3_COM_SEND_FIFO_SIZE
23	3_COM_SEND_FIFO_SIZE
24	3_COM_SEND_FIFO_SIZE
25	3_COM_SEND_FIFO_SIZE
26	3_COM_SEND_FIFO_SIZE
27	3_COM_SEND_FIFO_SIZE
28	3_COM_SEND_FIFO_SIZE
29	3_COM_SEND_FIFO_SIZE
30	3_COM_SEND_FIFO_SIZE
31	3_COM_SEND_FIFO_SIZE

3_COM_RCV_FIFO_SIZE RO, CON203[15..0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_RCV_FIFO_SIZE

3_COM_SEND_FIFO_SIZE RO, CON203[31..16], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_SEND_FIFO_SIZE

10.63 CON204

Bit	Name
0	3_COM_SEND_FIFO_CLR
1	3_COM_SEND_GO
2	Reserved
3	3_COM_CLR_SENT_CNT
4	3_COM_SENT_CNT
5	3_COM_SENT_CNT
6	3_COM_SENT_CNT
7	3_COM_SENT_CNT
8	3_COM_SENT_CNT
9	3_COM_SENT_CNT
10	3_COM_SENT_CNT
11	3_COM_SENT_CNT
12	3_COM_UP_SPD_MANUAL
13	3_COM_UP_SPD
14	Reserved
15	Reserved
16	3_COM_SEND_FIFO_CNT
17	3_COM_SEND_FIFO_CNT
18	3_COM_SEND_FIFO_CNT
19	3_COM_SEND_FIFO_CNT
20	3_COM_SEND_FIFO_CNT
21	3_COM_SEND_FIFO_CNT
22	3_COM_SEND_FIFO_CNT
23	3_COM_SEND_FIFO_CNT
24	3_COM_SEND_FIFO_CNT
25	3_COM_SEND_FIFO_CNT
26	3_COM_SEND_FIFO_CNT
27	3_COM_SEND_FIFO_CNT
28	3_COM_SEND_FIFO_CNT
29	3_COM_SEND_FIFO_CNT
30	3_COM_SEND_FIFO_CNT
31	3_COM_SEND_FIFO_CNT

3_COM_SEND_FIFO_CLR WO, CON204[0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_SEND_FIFO_CLR

3_COM_SEND_GO WO, CON204[1], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_SEND_GO

3_COM_CLR_SENT_CNT WO, CON204[3], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_CLR_SENT_CNT

3_COM_SENT_CNT RO, CON204[11..4], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_SENT_CNT.

3_COM_UP_SPD_MANUAL R/W, CON204[12], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_UP_SPD_MANUAL.

3_COM_UP_SPD R/W, CON204[13], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_UP_SPD.

3_COM_SEND_FIFO_CNT RO, CON204[31..16], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_SEND_FIFO_CNT

10.64 CON205

Bit	Name
0	3_COM_SEND_DATA
1	3_COM_SEND_DATA
2	3_COM_SEND_DATA
3	3_COM_SEND_DATA
4	3_COM_SEND_DATA
5	3_COM_SEND_DATA
6	3_COM_SEND_DATA
7	3_COM_SEND_DATA
8	3_COM_SEND_DATA
9	3_COM_SEND_DATA
10	3_COM_SEND_DATA
11	3_COM_SEND_DATA
12	3_COM_SEND_DATA
13	3_COM_SEND_DATA
14	3_COM_SEND_DATA
15	3_COM_SEND_DATA
16	3_COM_SEND_DATA
17	3_COM_SEND_DATA
18	3_COM_SEND_DATA
19	3_COM_SEND_DATA
20	3_COM_SEND_DATA
21	3_COM_SEND_DATA
22	3_COM_SEND_DATA
23	3_COM_SEND_DATA
24	3_COM_SEND_DATA
25	3_COM_SEND_DATA
26	3_COM_SEND_DATA
27	3_COM_SEND_DATA
28	3_COM_SEND_DATA
29	3_COM_SEND_DATA
30	3_COM_SEND_DATA
31	3_COM_SEND_DATA

3_COM_SEND_DATA R/W, CON205[31..0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_SEND_DATA

10.65 CON206

Bit	Name
0	3_COM_RCV_FIFO_CLR
1	Reserved
2	Reserved
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	3_COM_RCV_FIFO_CNT
17	3_COM_RCV_FIFO_CNT
18	3_COM_RCV_FIFO_CNT
19	3_COM_RCV_FIFO_CNT
20	3_COM_RCV_FIFO_CNT
21	3_COM_RCV_FIFO_CNT
22	3_COM_RCV_FIFO_CNT
23	3_COM_RCV_FIFO_CNT
24	3_COM_RCV_FIFO_CNT
25	3_COM_RCV_FIFO_CNT
26	3_COM_RCV_FIFO_CNT
27	3_COM_RCV_FIFO_CNT
28	3_COM_RCV_FIFO_CNT
29	3_COM_RCV_FIFO_CNT
30	3_COM_RCV_FIFO_CNT
31	3_COM_RCV_FIFO_CNT

3_COM_RCV_FIFO_CLR WO, CON206[0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_RCV_FIFO_CLR

3_COM_RCV_FIFO_CNT RO, CON206[31..16], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_RCV_FIFO_CNT

10.66 CON207

Bit	Name
0	3_COM_RCV_DATA
1	3_COM_RCV_DATA
2	3_COM_RCV_DATA
3	3_COM_RCV_DATA
4	3_COM_RCV_DATA
5	3_COM_RCV_DATA
6	3_COM_RCV_DATA
7	3_COM_RCV_DATA
8	3_COM_RCV_DATA
9	3_COM_RCV_DATA
10	3_COM_RCV_DATA
11	3_COM_RCV_DATA
12	3_COM_RCV_DATA
13	3_COM_RCV_DATA
14	3_COM_RCV_DATA
15	3_COM_RCV_DATA
16	3_COM_RCV_DATA
17	3_COM_RCV_DATA
18	3_COM_RCV_DATA
19	3_COM_RCV_DATA
20	3_COM_RCV_DATA
21	3_COM_RCV_DATA
22	3_COM_RCV_DATA
23	3_COM_RCV_DATA
24	3_COM_RCV_DATA
25	3_COM_RCV_DATA
26	3_COM_RCV_DATA
27	3_COM_RCV_DATA
28	3_COM_RCV_DATA
29	3_COM_RCV_DATA
30	3_COM_RCV_DATA
31	3_COM_RCV_DATA

3_COM_RCV_DATA RO, CON207[31..0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_COM_RCV_DATA

10.67 CON211

Bit	Name
0	3_LINK_INT_DEST
1	3_LINK_INT_DEST
2	Reserved
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

3_LINK_INT_DEST R/W, CON211[1..0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_LINK_INT_DEST

10.68 CON212

Bit	Name
0	3_UNDER_CURRENT
1	3_OVER_CURRENT
2	3_TRIG_ACK_RCVD
3	3_CTL_ACK_VER_ERR
4	3_CTL_ACK_RCVD
5	3_EVENT_RCVD
6	3_EVENT_FIFO_OVERFLOW
7	3_CTL_RSP_FIFO_OVF
8	3_CTL_REQ_FIFO_OVF
9	3_DOWN_TRIG_RCVD
10	3_TRIG_NOMATCH
11	3_IOACK_UNKNOWN_TYPE
12	3_IOACK_NOMATCH
13	3_HB_ERROR
14	3_HB_RCVD
15	3_STRM_PKT_DROP
16	3_STRM_NOT_ENOUGH_DAT
17	3_STRM_TOO_MUCH_DAT
18	3_STRM_BAD_CRC
19	3_STRM_OVERFLOW
20	3_STRM_CORNER
21	3_SERDES_LOST_ALIGN
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

3_UNDER_CURRENT R/W, CON212[0], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Undercurrent detected by POCXP controller interrupt.

3_OVER_CURRENT R/W, CON212[1], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Overcurrent detected by POCXP controller interrupt.

3_TRIG_ACK_RCVD R/W, CON212[2], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Trigger acknowledgment received from device interrupt.

3_CTL_ACK_VER_ERR R/W, CON212[3], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

There is a version mismatch between an incoming control packet acknowledgment and the board's current version settings. For example, the board could be in CXP 2.0 mode and it receives a CXP 1.1 acknowledgment, then this interrupt would be set.

3_CTL_ACK_RCVD R/W, CON212[4], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Control packet acknowledgment received from device interrupt.

3_EVENT_RCVD R/W, CON212[5], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

CXP 2.0 Event has been received.

3_EVENT_FIFO_OVERFLOW R/W, CON212[6], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

The CXP 2.0 Event FIFO has overflowed.

3_CTL_RSP_FIFO_OVF R/W, CON212[7], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Overflow detected in the control response FIFO (Device to host direction) interrupt.

3_CTL_REQ_FIFO_OVF R/W, CON212[8], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Overflow detected in control request FIFO (Host to device direction) interrupt.

3_DOWN_TRIG_RCVD R/W, CON212[9], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Received CXP trigger from camera.

3_TRIG_NOMATCH R/W, CON212[10], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Problem decoding a trigger packet from device interrupt.

3_IOACK_UNKNOWN_TYPE R/W, CON212[11], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Problem decoding an IO acknowledgment from device interrupt.

3_IOACK_NOMATCH R/W, CON212[12], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Problem decoding an IO acknowledgment from device interrupt.

3_HB_ERROR R/W, CON212[13], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

There is an error decoding a CXP 2.0 Heart Beat Packet.

3_HB_RCVD R/W, CON212[14], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

A CXP 2.0 Heart Beat packet has been received.

3_STRM_PKT_DROP R/W, CON212[15], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Problem decoding a stream packet header, remainder of packet is dropped interrupt.

3_STRM_NOT_ENOUGH_DAT R/W, CON212[16], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Not implemented, reserved interrupt.

3_STRM_TOO_MUCH_DAT R/W, CON212[17], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Not implemented, reserved interrupt.

3_STRM_BAD_CRC R/W, CON212[18], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

CRC error detected in stream packet interrupt.

3_STRM_OVERFLOW R/W, CON212[19], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Overflow in the stream packet buffer interrupt.

3_STRM_ CORNER R/W, CON212[19], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Not implemented, reserved interrupt.

3_SERDES_ LOST_ALIGN R/W, CON212[21], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

SERDES lost alignment interrupt.

10.69 CON213

Bit	Name
0	3_UNDER_CURRENT_M
1	3_OVER_CURRENT_M
2	3_TRIG_ACK_RCVD_M
3	3_GPIO_ACK_RCVD_M
4	3_CTL_ACK_RCVD_M
5	3_GPIO_RCVD_M
6	3_TRIG_RCVD_M
7	3_CTL_RSP_FIFO_OVF_M
8	3_CTL_REQ_FIFO_OVF_M
9	3_DOWN_TRIG_RCVD_M
10	3_TRIG_NOMATCH_M
11	3_IOACK_UNKNOWN_TYPE_M
12	3_IOACK_NOMATCH_M
13	3_IOACK_UNEXPECTED_INT_M
14	3_IOACK_NOMATCH2_M
15	3_STRM_PKT_DROP_M
16	3_STRM_NOT_ENOUGH_DAT_M
17	3_STRM_TOO_MUCH_DAT_M
18	3_STRM_BAD_CRC_M
19	3_STRM_OVERFLOW_M
20	3_STRM_CORNER_M
21	3_SERDES_LOST_ALIGN_M
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

3_UNDER_CURRENT_M R/W, CON213[0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_UNDER_CURRENT_M

3_OVER_CURRENT_M R/W, CON213[1], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_OVER_CURRENT_M

3_TRIG_ACK_RCVD_M R/W, CON213[2], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_TRIG_ACK_RCVD_M

3_GPIO_ACK_RCVD_M R/W, CON213[3], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_GPIO_ACK_RCVD_M

3_CTL_ACK_RCVD_M R/W, CON213[4], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_CTL_ACK_RCVD_M

3_GPIO_RCVD_M R/W, CON213[5], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_GPIO_RCVD_M

3_TRIG_RCVD_M R/W, CON213[6], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_TRIG_RCVD_M

3_CTL_RSP_FIFO_OVF_M R/W, CON213[7], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_CTL_RSP_FIFO_OVF_M

3_CTL_REQ_FIFO_OVF_M R/W, CON213[8], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_CTL_REQ_FIFO_OVF_M

3_DOWN_TRIG_RCVD_M R/W, CON213[9], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_DOWN_TRIG_RCVD_M

3_TRIG_NOMATCH_M R/W, CON213[10], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_TRIG_NOMATCH_M

3_IOACK_UNKNOWN_TYPE_M R/W, CON213[11], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_IOACK_UNKNOWN_TYPE_M

3_IOACK_NOMATCH_M R/W, CON213[12], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_IOACK_NOMATCH_M

3_IOACK_UNEXPECTED_INT_M R/W, CON213[13], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_IOACK_UNEXPECTED_INT_M

3_IOACK_NOMATCH2_M R/W, CON213[14], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_IOACK_NOMATCH2_M

3_STRM_PKT_DROP_M R/W, CON213[15], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_STRM_PKT_DROP_M

3_STRM_NOT_ENOUGH_DAT_M R/W, CON213[16], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_STRM_NOT_ENOUGH_DAT_M

3_STRM_TOO_MUCH_DAT_M R/W, CON213[17], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_STRM_TOO_MUCH_DAT_M

3_STRM_BAD_CRC_M R/W, CON213[18], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_STRM_BAD_CRC_M

3_STRM_OVERFLOW_M R/W, CON213[19], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_STRM_OVERFLOW_M

10.70 CON214

Bit	Name
0	3_UNDER_CURRENT_WP
1	3_OVER_CURRENT_WP
2	3_TRIG_ACK_RCVD_WP
3	3_GPIO_ACK_RCVD_WP
4	3_CTL_ACK_RCVD_WP
5	3_GPIO_RCVD_WP
6	3_TRIG_RCVD_WP
7	3_CTL_RSP_FIFO_OVF_WP
8	3_CTL_REQ_FIFO_OVF_WP
9	3_DOWN_TRIG_RCVD_WP
10	3_TRIG_NOMATCH_WP
11	3_IOACK_UNKNOWN_TYPE_WP
12	3_IOACK_NOMATCH_WP
13	3_IOACK_UNEXPECTED_INT_WP
14	3_IOACK_NOMATCH2_WP
15	3_STRM_PKT_DROP_WP
16	3_STRM_NOT_ENOUGH_DAT_WP
17	3_STRM_TOO_WPUCH_DAT_WP
18	3_STRM_BAD_CRC_WP
19	3_STRM_OVERFLOW_WP
20	3_STRM_CORNER_WP
21	3_SERDES_LOST_ALIGN_WP
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

3_UNDER_CURRENT_WP R/W, CON214[0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_UNDER_CURRENT_WP

3_OVER_CURRENT_WP R/W, CON214[1], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_OVER_CURRENT_WP

3_TRIG_ACK_RCVD_WP R/W, CON214[2], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_TRIG_ACK_RCVD_WP

3_GPIO_ACK_RCVD_WP R/W, CON214[3], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_GPIO_ACK_RCVD_WP

3_CTL_ACK_RCVD_WP R/W, CON214[4], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_CTL_ACK_RCVD_WP

3_GPIO_RCVD_WP R/W, CON214[5], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_GPIO_RCVD_WP

3_TRIG_RCVD_WP R/W, CON214[6], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_TRIG_RCVD_WP

3_CTL_RSP_FIFO_OVF_WP R/W, CON214[7], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_CTL_RSP_FIFO_OVF_WP

3_CTL_REQ_FIFO_OVF_WP R/W, CON214[8], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_CTL_REQ_FIFO_OVF_WP

3_DOWN_TRIG_RCVD_WP R/W, CON214[9], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_DOWN_TRIG_RCVD_WP

3_TRIG_NOMATCH_WP R/W, CON214[10], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_TRIG_NOMATCH_WP

3_IOACK_UNKNOWN_TYPE_WP R/W, CON214[11], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_IOACK_UNKNOWN_TYPE_WP

3_IOACK_NOMATCH_WP R/W, CON214[12], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_IOACK_NOMATCH_WP

3_IOACK_UNEXPECTED_INT_WP R/W, CON214[13], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_IOACK_UNEXPECTED_INT_WP

3_IOACK_NOMATCH2_WP R/W, CON214[14], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_IOACK_NOMATCH2_WP

3_STRM_PKT_DROP_WP R/W, CON214[15], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_STRM_PKT_DROP_WP

3_STRM_NOT_ENOUGH_DAT_WP R/W, CON214[16], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_STRM_NOT_ENOUGH_DAT_WP

3_STRM_TOO_WPUCH_DAT_WP R/W, CON214[17], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_STRM_TOO_WPUCH_DAT_WP

3_STRM_BAD_CRC_WP R/W, CON214[18], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_STRM_BAD_CRC_WP

3_STRM_OVERFLOW_WP R/W, CON214[19], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_STRM_OVERFLOW_WP

10.71 CON216

Bit	Name
0	3_PKT_RCVD_CNT
1	3_PKT_RCVD_CNT
2	3_PKT_RCVD_CNT
3	3_PKT_RCVD_CNT
4	3_PKT_RCVD_CNT
5	3_PKT_RCVD_CNT
6	3_PKT_RCVD_CNT
7	3_PKT_RCVD_CNT
8	3_PKT_RCVD_CNT
9	3_PKT_RCVD_CNT
10	3_PKT_RCVD_CNT
11	3_PKT_RCVD_CNT
12	3_PKT_RCVD_CNT
13	3_PKT_RCVD_CNT
14	3_PKT_RCVD_CNT
15	3_PKT_RCVD_CNT
16	3_PKT_GNT_CNT
17	3_PKT_GNT_CNT
18	3_PKT_GNT_CNT
19	3_PKT_GNT_CNT
20	3_PKT_GNT_CNT
21	3_PKT_GNT_CNT
22	3_PKT_GNT_CNT
23	3_PKT_GNT_CNT
24	3_PKT_GNT_CNT
25	3_PKT_GNT_CNT
26	3_PKT_GNT_CNT
27	3_PKT_GNT_CNT
28	3_PKT_GNT_CNT
29	3_PKT_GNT_CNT
30	3_PKT_GNT_CNT
31	3_PKT_GNT_CNT

3_PKT_RCVD_CNT RO, CON216[15..0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_PKT_RCVD_CNT

3_PKT_GNT_CNT RO, CON216[31..16], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_PKT_GNT_CNT

10.72 CON217

Bit	Name
0	3_PKT_DROP_CNT
1	3_PKT_DROP_CNT
2	3_PKT_DROP_CNT
3	3_PKT_DROP_CNT
4	3_PKT_DROP_CNT
5	3_PKT_DROP_CNT
6	3_PKT_DROP_CNT
7	3_PKT_DROP_CNT
8	3_PKT_DROP_CNT
9	3_PKT_DROP_CNT
10	3_PKT_DROP_CNT
11	3_PKT_DROP_CNT
12	3_PKT_DROP_CNT
13	3_PKT_DROP_CNT
14	3_PKT_DROP_CNT
15	3_PKT_DROP_CNT
16	3_CRC_ERR_CNT
17	3_CRC_ERR_CNT
18	3_CRC_ERR_CNT
19	3_CRC_ERR_CNT
20	3_CRC_ERR_CNT
21	3_CRC_ERR_CNT
22	3_CRC_ERR_CNT
23	3_CRC_ERR_CNT
24	3_CRC_ERR_CNT
25	3_CRC_ERR_CNT
26	3_CRC_ERR_CNT
27	3_CRC_ERR_CNT
28	3_CRC_ERR_CNT
29	3_CRC_ERR_CNT
30	3_CRC_ERR_CNT
31	3_CRC_ERR_CNT

3_PKT_DROP_CNT RO, CON217[15..0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_PKT_DROP_CNT

3_CRC_ERR_CNT RO, CON217[31..16], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_CRC_ERR_CNT

10.73 CON218

Bit	Name
0	3_CXP_TRIG_FALL_SENT
1	3_CXP_TRIG_FALL_SENT
2	3_CXP_TRIG_FALL_SENT
3	3_CXP_TRIG_FALL_SENT
4	3_CXP_TRIG_FALL_SENT
5	3_CXP_TRIG_FALL_SENT
6	3_CXP_TRIG_FALL_SENT
7	3_CXP_TRIG_FALL_SENT
8	3_CXP_TRIG_RISE_SENT
9	3_CXP_TRIG_RISE_SENT
10	3_CXP_TRIG_RISE_SENT
11	3_CXP_TRIG_RISE_SENT
12	3_CXP_TRIG_RISE_SENT
13	3_CXP_TRIG_RISE_SENT
14	3_CXP_TRIG_RISE_SENT
15	3_CXP_TRIG_RISE_SENT
16	3_CXP_TRIG_ACK_CNT
17	3_CXP_TRIG_ACK_CNT
18	3_CXP_TRIG_ACK_CNT
19	3_CXP_TRIG_ACK_CNT
20	3_CXP_TRIG_ACK_CNT
21	3_CXP_TRIG_ACK_CNT
22	3_CXP_TRIG_ACK_CNT
23	3_CXP_TRIG_ACK_CNT
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	3_CXP_TRIG_STATE

3_CXP_TRIG_FALL_SENT RO, CON218[7..0], Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of CXP trigger falling edge trigger packets sent to the camera.

3_CXP_TRIG_RISE_SENT RO, CON218[15..8], Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of CXP trigger rising edge trigger packets sent to the camera.

3_CXP_TRIG_ACK_CNT RO, CON218[23..16], Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of CXP trigger acknowledgments received from the camera.

3_CXP_TRIG_STATE RO, CON218[31], Claxon-CXP, Claxon-FXP, Cyton-CXP

Current state of the uplink CXP trigger signal.

10.74 CON218

Bit	Name
0	3_CXP_TRIG_STATE
1	Reserved
2	Reserved
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	3_CXP_TRIG_ACK_CNT
25	3_CXP_TRIG_ACK_CNT
26	3_CXP_TRIG_ACK_CNT
27	3_CXP_TRIG_ACK_CNT
28	3_CXP_TRIG_ACK_CNT
29	3_CXP_TRIG_ACK_CNT
30	3_CXP_TRIG_ACK_CNT
31	3_CXP_TRIG_ACK_CNT

3_CXP_TRIG_STATE RO, CON218[0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_CXP_TRIG_STATE

3_CXP_TRIG_ACK_CNT RO, CON218[31..24], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_CXP_TRIG_ACK_CNT

10.75 CON219

Bit	Name
0	3_DT_FALL_RCVD_COUNT
1	3_DT_FALL_RCVD_COUNT
2	3_DT_FALL_RCVD_COUNT
3	3_DT_FALL_RCVD_COUNT
4	3_DT_FALL_RCVD_COUNT
5	3_DT_FALL_RCVD_COUNT
6	3_DT_FALL_RCVD_COUNT
7	3_DT_FALL_RCVD_COUNT
8	3_DT_RISE_RCVD_COUNT
9	3_DT_RISE_RCVD_COUNT
10	3_DT_RISE_RCVD_COUNT
11	3_DT_RISE_RCVD_COUNT
12	3_DT_RISE_RCVD_COUNT
13	3_DT_RISE_RCVD_COUNT
14	3_DT_RISE_RCVD_COUNT
15	3_DT_RISE_RCVD_COUNT
16	3_DT_TRIG_DELAY
17	3_DT_TRIG_DELAY
18	3_DT_TRIG_DELAY
19	3_DT_TRIG_DELAY
20	3_DT_TRIG_DELAY
21	3_DT_TRIG_DELAY
22	3_DT_TRIG_DELAY
23	3_DT_TRIG_DELAY
24	3_DT_FALL_DISABLE
25	3_DT_RISE_DISABLE
26	3_DT_TRIG_NUM
27	3_DT_TRIG_NUM
28	3_DT_TRIG_NUM
29	3_DT_TRIG_NUM
30	3_DT_CLR_CNT
31	3_DT_STATE

3_DT_FALL_RCVD_COUNT RO, CON219[7..0], Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of CXP down stream (camera to frame grabber) trigger falling edges.

3_DT_RISE_RCVD_COUNT RO, CON219[15..8], Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of CXP down stream (camera to frame grabber) trigger rising edges.

3_DT_TRIG_DELAY RO, CON219[23..16], Claxon-CXP, Claxon-FXP, Cyton-CXP

The payload of the CXP down stream (camera to frame grabber) trigger.

3_DT_FALL_DISABLE R/W, CON219[24], Claxon-CXP, Claxon-FXP, Cyton-CXP

Setting this bit to 1 will disable the falling edge trigger packet from causing an interrupt. It will still be counted and affect DT_STATE.

3_DT_RISE_DISABLE R/W, CON219[25], Claxon-CXP, Claxon-FXP, Cyton-CXP

Setting this bit to 1 will disable the rising edge trigger packet from causing an interrupt. It will still be counted and affect DT_STATE.

3_DT_TRIG_NUM RO, CON219[29..26], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

When the link is in CXP 2.0 or later mode, these bits capture the TriggerN field of a trigger.

3_DT_CLR_CNT WO, CON219[30], Claxon-CXP, Claxon-FXP, Cyton-CXP

Writing this register to 1 will clear that counters in this registers.

3_DT_STATE RO, CON219[31], Claxon-CXP, Claxon-FXP, Cyton-CXP

Current stated of the downstream trigger.

10.76 CON223

Bit	Name
0	3_SERDES_STATE
1	3_SERDES_STATE
2	3_SERDES_STATE
3	3_SERDES_STATE
4	3_SERDES_STATE
5	3_SERDES_STATE
6	3_SERDES_STATE
7	3_SERDES_STATE
8	3_LINK_SPEED
9	3_LINK_SPEED
10	3_LINK_SPEED
11	3_LINK_SPEED
12	3_LINK_SPEED
13	3_LINK_SPEED
14	3_LINK_SPEED
15	3_LINK_SPEED
16	3_LOST_ALIGN_CNT
17	3_LOST_ALIGN_CNT
18	3_LOST_ALIGN_CNT
19	3_LOST_ALIGN_CNT
20	3_LOST_ALIGN_CNT
21	3_LOST_ALIGN_CNT
22	3_LOST_ALIGN_CNT
23	3_LOST_ALIGN_CNT
24	3_LOST_ALIGN_CNT
25	3_LOST_ALIGN_CNT
26	Reserved
27	Reserved
28	3_RESYNC_TOG_MSTR
29	3_RESYNC_TOG
30	3_SERDES_ALIGNED
31	Reserved

3_SERDES_STATE RO, CON223[7..0], Claxon-CXP, Claxon-FXP, Cyton-CXP

Current state of the SERDES associated with this link.

3_LINK_SPEED RO, CON223[15..8], Claxon-CXP, Claxon-FXP, Cyton-CXP

Current speed of the SERDES associated with this link.

3_LOST_ALIGN_CNT RO, CON223[25..16], Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of times the SERDES associated with this link has lost alignment.

3_RESYNC_TOG_MSTR RO, CON223[28], Claxon-CXP, Claxon-FXP, Cyton-CXP

Diagnostics.

3_RESYNC_TOG RO, CON223[29], Claxon-CXP, Claxon-FXP, Cyton-CXP

Diagnostics.

3_SERDES_ALIGNED RO, CON223[30], Claxon-CXP, Claxon-FXP, Cyton-CXP

If this bit is 1 the SERDES associated with this link is aligned.

10.77 CON224

Bit	Name
0	3_RAW_DATA_MODE
1	3_REMOVE_IDLES
2	3_DISABLE_SUB
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	3_MAX_PKT_RCVD
17	3_MAX_PKT_RCVD
18	3_MAX_PKT_RCVD
19	3_MAX_PKT_RCVD
20	3_MAX_PKT_RCVD
21	3_MAX_PKT_RCVD
22	3_MAX_PKT_RCVD
23	3_MAX_PKT_RCVD
24	3_MAX_PKT_RCVD
25	3_MAX_PKT_RCVD
26	3_MAX_PKT_RCVD
27	3_MAX_PKT_RCVD
28	3_MAX_PKT_RCVD
29	3_MAX_PKT_RCVD
30	3_MAX_PKT_RCVD
31	3_MAX_PKT_RCVD

3_RAW_DATA_MODE R/W, CON224[0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_RAW_DATA_MODE

3_REMOVE_IDLES R/W, CON224[1], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_REMOVE_IDLES

3_DISABLE_SUB R/W, CON224[2], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

This bit controls the automatic pixel substitution facility built into the board's receiver circuit. In some very unusual circumstance that low level CXP hardware and perform a pixel substitution if bad data packet is detected. This substitution will still result in a CRC error for the packet, but it will cause less harm than if the original bad packet was received. Setting this bit to 1 will turn this facility off, which is really only useful for testing.

3_MAX_PKT_RCVD R/W, CON224[16], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

The maximum CXP packet that has been received from the camera. Units are 32-bit words. Clear this register by writing a 0 to it.

10.78 CON227

Bit	Name
0	3_SERDES_ERROR_CODE
1	3_SERDES_ERROR_CODE
2	3_SERDES_ERROR_CODE
3	3_SERDES_ERROR_CODE
4	3_SERDES_ERROR_CODE
5	3_SERDES_ERROR_CODE
6	3_SERDES_ERROR_CODE
7	3_SERDES_ERROR_CODE
8	3_SERDES_ERROR_CODE
9	3_SERDES_ERROR_CODE
10	3_SERDES_ERROR_CODE
11	3_SERDES_ERROR_CODE
12	3_SERDES_ERROR_CODE
13	3_SERDES_ERROR_CODE
14	3_SERDES_ERROR_CODE
15	3_SERDES_ERROR_CODE
16	3_SERDES_ERROR_CODE
17	3_SERDES_ERROR_CODE
18	3_SERDES_ERROR_CODE
19	3_SERDES_ERROR_CODE
20	3_SERDES_ERROR_CODE
21	3_SERDES_ERROR_CODE
22	3_SERDES_ERROR_CODE
23	3_SERDES_ERROR_CODE
24	3_SERDES_ERROR_CODE
25	3_SERDES_ERROR_CODE
26	3_SERDES_ERROR_CODE
27	3_SERDES_ERROR_CODE
28	3_SERDES_ERROR_CODE
29	3_SERDES_ERROR_CODE
30	3_SERDES_ERROR_CODE
31	Reserved

3_SERDES_ERROR_CODE RO, CON227[30..0], Claxon-CXP, Claxon-FXP, Cyton-CXP

See description of 0_SERDES_ERROR_CODE

10.79 CON356

Bit	Name
0	FW_BUILD_YEAR
1	FW_BUILD_YEAR
2	FW_BUILD_YEAR
3	FW_BUILD_YEAR
4	FW_BUILD_YEAR
5	FW_BUILD_YEAR
6	FW_BUILD_YEAR
7	FW_BUILD_YEAR
8	FW_BUILD_YEAR
9	FW_BUILD_YEAR
10	FW_BUILD_YEAR
11	FW_BUILD_YEAR
12	FW_BUILD_YEAR
13	FW_BUILD_YEAR
14	FW_BUILD_YEAR
15	Reserved
16	FW_BUILD_DAY
17	FW_BUILD_DAY
18	FW_BUILD_DAY
19	FW_BUILD_DAY
20	FW_BUILD_DAY
21	FW_BUILD_DAY
22	FW_BUILD_DAY
23	FW_BUILD_DAY
24	FW_BUILD_MONTH
25	FW_BUILD_MONTH
26	FW_BUILD_MONTH
27	FW_BUILD_MONTH
28	FW_BUILD_MONTH
29	FW_BUILD_MONTH
30	FW_BUILD_MONTH
31	FW_BUILD_MONTH

FW_BUILD_YEAR RO, CON356[15..0], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Year that this firmware was compiled in BCD format. Example: 0x2012 is year 2012

FW_BUILD_DAY RO, CON356[23..16], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Day that this firmware was compiled in BCD format. Example: 0x18 the 18th of the month.

FW_BUILD_MONTH RO, CON356[31..24], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Month that this firmware was compiled in BCD format. Example: 0x12 is december.

10.80 CON357

Bit	Name
0	FW_BUILD_MIN
1	FW_BUILD_MIN
2	FW_BUILD_MIN
3	FW_BUILD_MIN
4	FW_BUILD_MIN
5	FW_BUILD_MIN
6	FW_BUILD_MIN
7	FW_BUILD_MIN
8	FW_BUILD_HOUR
9	FW_BUILD_HOUR
10	FW_BUILD_HOUR
11	FW_BUILD_HOUR
12	FW_BUILD_HOUR
13	FW_BUILD_HOUR
14	FW_BUILD_HOUR
15	FW_BUILD_HOUR
16	FPGA_ID
17	FPGA_ID
18	FPGA_ID
19	FPGA_ID
20	FPGA_ID
21	FPGA_ID
22	FPGA_ID
23	FPGA_ID
24	FW_CMPTBL
25	FW_CMPTBL
26	FW_CMPTBL
27	FW_CMPTBL
28	FW_CMPTBL
29	FW_CMPTBL
30	FW_CMPTBL
31	FW_CMPTBL

FW_BUILD_MIN RO, CON357[7..0], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Minute that this firmware was compiled. Example: 0x35 is 35 minutes past the hour.

FW_BUILD_HOUR RO, CON357[15..8], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Hour that this firmware was compiled. Example: 0x23 is 11pm (23rd hour).

FPGA_ID RO, CON357[23..16], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

FPGA Identifier

FW_CMPTBL RO, CON357[31..24], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Firmware compatibility version (must match SDK driver internal firmware version).

10.81 CON358

Bit	Name
0	PFG_RESET
1	Reserved
2	Reserved
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	FPGA_TEMP
17	FPGA_TEMP
18	FPGA_TEMP
19	FPGA_TEMP
20	FPGA_TEMP
21	FPGA_TEMP
22	FPGA_TEMP
23	FPGA_TEMP
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

PFG_RESET R/W, CON358[0], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Reset the PFG (Stream assembler). This register does not get cleared automatically. Software must clear it.

FPGA_TEMP RO, CON358[23..16], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

FPGA die temperature in degrees Celsius. Value is two complement number, i.e. temperature can go negative.

10.82 CON359

Bit	Name
0	CXP_VER_SUP
1	CXP_VER_SUP
2	CXP_VER_SUP
3	CXP_VER_SUP
4	CXP_VER_SUP
5	CXP_VER_SUP
6	CXP_VER_SUP
7	CXP_VER_SUP
8	CXP_VER_SUP
9	CXP_VER_SUP
10	CXP_VER_SUP
11	CXP_VER_SUP
12	CXP_VER_SUP
13	CXP_VER_SUP
14	CXP_VER_SUP
15	CXP_VER_SUP
16	CXP_VER_SUP
17	CXP_VER_SUP
18	CXP_VER_SUP
19	CXP_VER_SUP
20	CXP_VER_SUP
21	CXP_VER_SUP
22	CXP_VER_SUP
23	CXP_VER_SUP
24	CXP_VER_SUP
25	CXP_VER_SUP
26	CXP_VER_SUP
27	CXP_VER_SUP
28	CXP_VER_SUP
29	CXP_VER_SUP
30	CXP_VER_SUP
31	CXP_VER_SUP

CXP_VER_SUP R/W, CON358[0], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

The register indicates the maximum CXP version the board currently supports. The board will also support older CXP versions.

CX_VER_SUP	Meaning
0x00010000	Supports CXP Version 1.0 or earlier
0x00010001	Supports CXP Version 1.1 or earlier
0x00020000	Supports CXP Version 2.0 or earlier

10.83 CON360

Bit	Name
0	NUM_LINKS_NEEDED
1	NUM_LINKS_NEEDED
2	NUM_LINKS_NEEDED
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

NUM_LINKS_NEEDED R/W, CON360[2..0], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of active CXP links that a Stream assembler will be working with. The LINK_ORDER register should be configured prior to setting NUM_LINKS to a non-zero value.

10.84 CON361

Bit	Name
0	LINK_0_ORDER
1	LINK_0_ORDER
2	LINK_0_ORDER
3	LINK_0_ORDER
4	LINK_1_ORDER
5	LINK_1_ORDER
6	LINK_1_ORDER
7	LINK_1_ORDER
8	LINK_2_ORDER
9	LINK_2_ORDER
10	LINK_2_ORDER
11	LINK_2_ORDER
12	LINK_3_ORDER
13	LINK_3_ORDER
14	LINK_3_ORDER
15	LINK_3_ORDER
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

LINK_0_ORDER R/W, CON361[3..0], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

First physical link that the stream decoder will accept packets from. NUM_LINKS register determines how many of these physical links will be considered in arbitration.

LINK_1_ORDER R/W, CON361[7..4], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Second physical link that the stream decoder will accept packets from. NUM_LINKS register determines how many of these physical links will be considered in arbitration.

LINK_2_ORDER R/W, CON361[11..8], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Third physical link that the stream decoder will accept packets from. NUM_LINKS register determines how many of these physical links will be considered in arbitration.

LINK_3_ORDER R/W, CON361[15..12], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Fourth physical link that the stream decoder will accept packets from. NUM_LINKS register determines how many of these physical links will be considered in arbitration.

10.85 CON363

Bit	Name
0	CXP_OUT_TABLE_EVEN
1	CXP_OUT_TABLE_EVEN
2	CXP_OUT_TABLE_EVEN
3	CXP_OUT_TABLE_EVEN
4	CXP_OUT_TABLE_ODD
5	CXP_OUT_TABLE_ODD
6	CXP_OUT_TABLE_ODD
7	CXP_OUT_TABLE_ODD
8	CXP_OUT_TABLE_LPC
9	CXP_OUT_TABLE_LPC
10	CXP_OUT_TABLE_LPC
11	CXP_OUT_TABLE_LPC
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	CXP_OUT_TABLE_EN

CXP_OUT_TABLE_EVEN R/W, CON363[3..0], Karbon-CXP

Selects which VFG should receive the even lines.

CXP_OUT_TABLE_ODD R/W, CON363[7..4], Karbon-CXP

Selects which VFG should receive the odd lines.

CXP_OUT_TABLE_LPC R/W, CON363[11..8], Karbon-CXP

Number of lines to send to an odd or even VFG channel before switching.

CXP_OUT_TABLE_EN R/W, CON363[31], Karbon-CXP

Enable the forwarding of video data to the DMA engines (VFGs)

10.86 CON372

Bit	Name
0	PKT_GNT_CNT
1	PKT_GNT_CNT
2	PKT_GNT_CNT
3	PKT_GNT_CNT
4	PKT_GNT_CNT
5	PKT_GNT_CNT
6	PKT_GNT_CNT
7	PKT_GNT_CNT
8	PKT_GNT_CNT
9	PKT_GNT_CNT
10	PKT_GNT_CNT
11	PKT_GNT_CNT
12	PKT_GNT_CNT
13	PKT_GNT_CNT
14	PKT_GNT_CNT
15	PKT_GNT_CNT
16	NEXT_LINK
17	NEXT_LINK
18	NEXT_LINK
19	DISABLE_PKT_TAG_CHK
20	UNEXP_PKT_TAG_CNT
21	UNEXP_PKT_TAG_CNT
22	UNEXP_PKT_TAG_CNT
23	UNEXP_PKT_TAG_CNT
24	UNEXP_PKT_TAG_CNT
25	UNEXP_PKT_TAG_CNT
26	UNEXP_PKT_TAG_CNT
27	UNEXP_PKT_TAG_CNT
28	Reserved
29	Reserved
30	Reserved
31	ACTIVE

PKT_GNT_CNT RO, CON372[15..0], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Total number of packets accepted from all links associated with this stream decoder. The counter wraps and is for debug only.

NEXT_LINK RO, CON372[18..16], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

The next link from which a packet is expected.

DISABLE_PKT_TAG_CHK RO, CON372[19], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Disable the checking of packet tags.

UNEXP_PKT_TAG_CNT RO, CON372[27..20], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Number of unexpected packet tags received. These are out-of-order tags. The counter wraps and is for debug only.

ACTIVE RO, CON372[31], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

The stream decoder is currently accepting a packet.

10.87 CON428

Bit	Name
0	FORCE_RESYNC
1	PACK_PIXELS
2	Reserved
3	Reserved
4	STRM_DECODE_STATE
5	STRM_DECODE_STATE
6	STRM_DECODE_STATE
7	Reserved
8	SOF_ERR
9	SOL_ERR
10	IH_ERR
11	SRC_TAG_ERR
12	IS_LINE_SCAN
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

FORCE_RESYNC WO, CON428[0], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Forces re-sync of CXP engine.

PACK_PIXELS WO, CON428[1], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Denotes that pixels will be DMAed packed.

STRM_DECODE_STATE RO, CON428[6..4], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Current state CXP stream decode engine.

SOF_ERR RO, CON428[8], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Bad Start Of Frame Packet.

SOL_ERR RO, CON428[9], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Bad Start Of Line Packet

IH_ERR RO, CON428[10], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Bad Image Header.

SRC_TAG_ERR RO, CON428[11], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Unexpected Source Tag error.

IS_LINE_SCAN RO, CON428[12], Aon-CXP, Claxon-CXP, Claxon-FXP, Cyton-CXP

Camera is a line scan camera.

Encoder Based Processing Control

Chapter 11

11.1 Introduction

The registers in this section configure a number of features that primarily use a connected encoder to control the flow trigger inputs, frame acquisition and output pulse events. The main use cases involve situations where there is a continuous flow of objects for inspection, and the Part In Place (PIP) trigger is not be located coincident with the camera, so a delay between the PIP and the start of acquisition is needed. To further complicate things, there may be multiple objects between the trigger and the camera, so there needs to be a way to track each trigger and trigger acquisition when the object gets below the camera. The solution here is to use quadrature encoder counter as a kind of timestamp. Which can be use to keep track of multiple objects and events

This idea extends to creating output events that need to take place in the future. Again the use of the encoder count as a time stamp is the solution. For example, image processing determines that a part is defective. The processing engine can put an “eject” pulse event into a FIFO along with the corresponding object’s timestamp. After some delay, once the object is adjacent to the actuator, it can push the part into a reject bin.

Then Encoder Based Process Control unit has two main modes:

- Encoder Based Trigger Delay - an external hardware trigger is delayed by a programmed amount of encoder pulses. Once that amount of encoder pulse has passed, the system is triggered. This mode has two sub modes: Automatic Hardware mode and Manual mode.
- Encoder Based Output Event Delay - output events are put into a FIFO to be executed at some point in the future. The events consist of 1 or more pulse on separate output pins of programmable duration.

11.2 CON74

Bit	Name
0	PR_TRIG_ENC_FIFO
1	PR_TRIG_ENC_FIFO
2	PR_TRIG_ENC_FIFO
3	PR_TRIG_ENC_FIFO
4	PR_TRIG_ENC_FIFO
5	PR_TRIG_ENC_FIFO
6	PR_TRIG_ENC_FIFO
7	PR_TRIG_ENC_FIFO
8	PR_TRIG_ENC_FIFO
9	PR_TRIG_ENC_FIFO
10	PR_TRIG_ENC_FIFO
11	PR_TRIG_ENC_FIFO
12	PR_TRIG_ENC_FIFO
13	PR_TRIG_ENC_FIFO
14	PR_TRIG_ENC_FIFO
15	PR_TRIG_ENC_FIFO
16	PR_TRIG_ENC_FIFO
17	PR_TRIG_ENC_FIFO
18	PR_TRIG_ENC_FIFO
19	PR_TRIG_ENC_FIFO
20	PR_TRIG_ENC_FIFO
21	PR_TRIG_ENC_FIFO
22	PR_TRIG_ENC_FIFO
23	PR_TRIG_ENC_FIFO
24	PR_TRIG_ENC_FIFO
25	PR_TRIG_ENC_FIFO
26	PR_TRIG_ENC_FIFO
27	PR_TRIG_ENC_FIFO
28	PR_TRIG_ENC_FIFO
29	PR_TRIG_ENC_FIFO
30	PR_TRIG_ENC_FIFO
31	PR_TRIG_ENC_FIFO

PR_TRIG_ENC_FIFO R/W, CON74[31..0], Aon, Axion, Claxon and Cyton

When the Encoder Based Trigger Delay system is in Manual mode (TRG_DLY_MODE = 1), reading this register pops the next value from the Triggered Encoder Count FIFO. This value will represent the encoder count of the next object that needs to be processed was triggered.

When TRG_DLY_MODE = 0, reading this register has no effect.

11.3 CON75

Bit	Name
0	PR_SOF_ENC_FIFO
1	PR_SOF_ENC_FIFO
2	PR_SOF_ENC_FIFO
3	PR_SOF_ENC_FIFO
4	PR_SOF_ENC_FIFO
5	PR_SOF_ENC_FIFO
6	PR_SOF_ENC_FIFO
7	PR_SOF_ENC_FIFO
8	PR_SOF_ENC_FIFO
9	PR_SOF_ENC_FIFO
10	PR_SOF_ENC_FIFO
11	PR_SOF_ENC_FIFO
12	PR_SOF_ENC_FIFO
13	PR_SOF_ENC_FIFO
14	PR_SOF_ENC_FIFO
15	PR_SOF_ENC_FIFO
16	PR_SOF_ENC_FIFO
17	PR_SOF_ENC_FIFO
18	PR_SOF_ENC_FIFO
19	PR_SOF_ENC_FIFO
20	PR_SOF_ENC_FIFO
21	PR_SOF_ENC_FIFO
22	PR_SOF_ENC_FIFO
23	PR_SOF_ENC_FIFO
24	PR_SOF_ENC_FIFO
25	PR_SOF_ENC_FIFO
26	PR_SOF_ENC_FIFO
27	PR_SOF_ENC_FIFO
28	PR_SOF_ENC_FIFO
29	PR_SOF_ENC_FIFO
30	PR_SOF_ENC_FIFO
31	PR_SOF_ENC_FIFO

PR_SOF_ENC_FIFO R/W, CON75[31..0], Aon, Axion, Claxon and Cyton

When the Encoder Based Trigger Delay system is in Manual mode (TRG_DLY_MODE = 1), reading this register pops the next value from the Start of Frame (SOF) FIFO. This value will represent the encoder count of the Start of Frame event of the next object that needs to be processed.

When TRG_DLY_MODE = 0, reading this register has no effect.

11.4 CON76

Bit	Name
0	PR_SOF_ENC_WORDS_USED
1	PR_SOF_ENC_WORDS_USED
2	PR_SOF_ENC_WORDS_USED
3	PR_SOF_ENC_WORDS_USED
4	PR_SOF_ENC_WORDS_USED
5	PR_SOF_ENC_WORDS_USED
6	PR_SOF_ENC_WORDS_USED
7	PR_SOF_ENC_WORDS_USED
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	PR_SOF_ENC_FIFO_CLR
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	PR_SOF_ENC_FIFO_OFLW

PR_SOF_ENC_ WORDS_USED RO, CON76[7..0], Aon, Axion, Claxon and Cyton

Number of entries in the Start of Frame FIFO.

PR_SOF_ENC_ FIFO_CLR R/W, CON76[23], Aon, Axion, Claxon and Cyton

Write a 1 to this register will clear the Start of Fame FIFO

PR_SOF_ENC_ FIFO_OFLW RO, CON76[31], Aon, Axion, Claxon and Cyton

If this bit reads back 1, there has been a FIFO overflow in the Start of Fame FIFO.

11.5 CON77

Bit	Name
0	RD_ENCODER_CNT
1	RD_ENCODER_CNT
2	RD_ENCODER_CNT
3	RD_ENCODER_CNT
4	RD_ENCODER_CNT
5	RD_ENCODER_CNT
6	RD_ENCODER_CNT
7	RD_ENCODER_CNT
8	RD_ENCODER_CNT
9	RD_ENCODER_CNT
10	RD_ENCODER_CNT
11	RD_ENCODER_CNT
12	RD_ENCODER_CNT
13	RD_ENCODER_CNT
14	RD_ENCODER_CNT
15	RD_ENCODER_CNT
16	RD_ENCODER_CNT
17	RD_ENCODER_CNT
18	RD_ENCODER_CNT
19	RD_ENCODER_CNT
20	RD_ENCODER_CNT
21	RD_ENCODER_CNT
22	RD_ENCODER_CNT
23	RD_ENCODER_CNT
24	RD_ENCODER_CNT
25	RD_ENCODER_CNT
26	RD_ENCODER_CNT
27	RD_ENCODER_CNT
28	RD_ENCODER_CNT
29	RD_ENCODER_CNT
30	RD_ENCODER_CNT
31	RD_ENCODER_CNT

RD_ENCODER_CNT RO, CON77[31..0], Aon, Axion, Claxon and Cyton

The current 32-bit value of the quadrature encoder counter.

11.6 CON78

Bit	Name
0	TRG_DLY_AMOUNT
1	TRG_DLY_AMOUNT
2	TRG_DLY_AMOUNT
3	TRG_DLY_AMOUNT
4	TRG_DLY_AMOUNT
5	TRG_DLY_AMOUNT
6	TRG_DLY_AMOUNT
7	TRG_DLY_AMOUNT
8	TRG_DLY_AMOUNT
9	TRG_DLY_AMOUNT
10	TRG_DLY_AMOUNT
11	TRG_DLY_AMOUNT
12	TRG_DLY_AMOUNT
13	TRG_DLY_AMOUNT
14	TRG_DLY_AMOUNT
15	TRG_DLY_AMOUNT
16	TRG_DLY_AMOUNT
17	TRG_DLY_AMOUNT
18	TRG_DLY_AMOUNT
19	TRG_DLY_AMOUNT
20	TRG_DLY_AMOUNT
21	TRG_DLY_AMOUNT
22	TRG_DLY_AMOUNT
23	TRG_DLY_AMOUNT
24	TRG_DLY_AMOUNT
25	TRG_DLY_AMOUNT
26	TRG_DLY_AMOUNT
27	TRG_DLY_AMOUNT
28	TRG_DLY_AMOUNT
29	TRG_DLY_AMOUNT
30	TRG_DLY_AMOUNT
31	TRG_DLY_AMOUNT

TRG_DLY_AMOUNT

R/W, CON78[31..0], Aon, Axion, Claxon and Cyton

When the system is in Encoder Based Trigger Delay mode, this register sets the amount of delay between actual trigger assertion and when board will start the acquisition of a new frame. The units here are encoder counts. To program this register correctly, you must determine the distance between the PIP sensor and the camera. Then convert this distance to quadrature encoder pulses. Then use this value to program this register.

11.7 CON79

Bit	Name
0	TRG_DLY_WORDS_USED
1	TRG_DLY_WORDS_USED
2	TRG_DLY_WORDS_USED
3	TRG_DLY_WORDS_USED
4	TRG_DLY_WORDS_USED
5	TRG_DLY_WORDS_USED
6	TRG_DLY_WORDS_USED
7	TRG_DLY_WORDS_USED
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	TRG_DLY_FIFO_CLR
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	TRG_DLY_FIFO_OFLW

TRG_DLY_WORDS_USED RO, CON79[7..0], Aon, Axion, Claxon and Cyton

Number of entries in the Triggered Encoder Count FIFO.

TRG_DLY_FIFO_CLR R/W, CON79[23], Aon, Axion, Claxon and Cyton

Writing this bit to 1 will clear that Triggered Encoder Count FIFO.

TRG_DLY_FIFO_OFLW RO, CON79[31], Aon, Axion, Claxon and Cyton

If this bit reads back 1, the Triggered Encoder Count FIFO has overflowed.

11.8 CON80

Bit	Name
0	TRG_DLY_EN
1	Reserved
2	TRG_DLY_TRIG_SEL
3	Reserved
4	TRG_DLY_SM_RST
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	TRG_DLY_MODE
29	TRG_DLY_MODE
30	TRG_DLY_MODE
31	TRG_DLY_MODE

TRG_DLY_EN R/W, CON80[0], Aon, Axion, Claxon and Cyton

Setting this bit to 1 enable Encoder Based Trigger Delay mode. When this bit is 0 the trigger works normally (i.e. immediately causes start of acquisition).

TRG_DLY_TRIG_SEL R/W, CON80[2], Aon, Axion, Claxon and Cyton

When the board is in Encoder Based Trigger Delay mode, this bit tells the system which type of camera is being used.

TRG_DLY_TRIG_SEL	Meaning
0	Line scan camera
1	Area scan camera

TRG_DLY_SM_RST R/W, CON80[4], Aon, Axion, Claxon and Cyton

This bit is use to reset the Encoder Based Trigger Delay system. Write this bit to 1, then write it back to 0 to reset the system.

TRG_DLY_MODE R/W, CON80[31..28], Aon, Axion, Claxon and Cyton

This bit set Encoder Based Trigger Delay system mode.

TRG_DLY_MODE	Meaning
0	Automatic Hardware mode: triggers are automatically pushed on the FIFO, then after the trigger delay amount has passed, the acquisition system is automatically trigger. No host interaction is required.
1	Manual mode: triggers are automatically pushed on the FIFO, but the host software must pop them off the FIFO manually. Usually in this mode acquisition is free-running, and the trigger is used to "timestamp" the encoder count where (when) the trigger occurred. The host software can use this information and the SOF FIFO to built virtual frames of any size and of any relationship to the trigger out of the lines that have been acquired into memory.

11.9 CON81

Bit	Name
0	TRG_DLY_CNT_LT
1	TRG_DLY_CNT_LT
2	TRG_DLY_CNT_LT
3	TRG_DLY_CNT_LT
4	TRG_DLY_CNT_LT
5	TRG_DLY_CNT_LT
6	TRG_DLY_CNT_LT
7	TRG_DLY_CNT_LT
8	TRG_DLY_CNT_LT
9	TRG_DLY_CNT_LT
10	TRG_DLY_CNT_LT
11	TRG_DLY_CNT_LT
12	TRG_DLY_CNT_LT
13	TRG_DLY_CNT_LT
14	TRG_DLY_CNT_LT
15	TRG_DLY_CNT_LT
16	TRG_DLY_CNT_LT
17	TRG_DLY_CNT_LT
18	TRG_DLY_CNT_LT
19	TRG_DLY_CNT_LT
20	TRG_DLY_CNT_LT
21	TRG_DLY_CNT_LT
22	TRG_DLY_CNT_LT
23	TRG_DLY_CNT_LT
24	TRG_DLY_CNT_LT
25	TRG_DLY_CNT_LT
26	TRG_DLY_CNT_LT
27	TRG_DLY_CNT_LT
28	TRG_DLY_CNT_LT
29	TRG_DLY_CNT_LT
30	TRG_DLY_CNT_LT
31	TRG_DLY_CNT_LT

TRG_DLY_CNT_LT RO, CON81[31..0], Aon, Axion, Claxon and Cyton

This is the value of the quadrature encoder counter at the end of an acquired frame. Since this value is latched, it can be ready any time up until the end of the next frame (at which point the value will be overwritten). This value should be equal to the trigger encoder count plus the trigger delay plus the frame size.

11.10 CON83

Bit	Name
0	OUT_DLY_EVENT_CNT
1	OUT_DLY_EVENT_CNT
2	OUT_DLY_EVENT_CNT
3	OUT_DLY_EVENT_CNT
4	OUT_DLY_EVENT_CNT
5	OUT_DLY_EVENT_CNT
6	OUT_DLY_EVENT_CNT
7	OUT_DLY_EVENT_CNT
8	OUT_DLY_EVENT_CNT
9	OUT_DLY_EVENT_CNT
10	OUT_DLY_EVENT_CNT
11	OUT_DLY_EVENT_CNT
12	OUT_DLY_EVENT_CNT
13	OUT_DLY_EVENT_CNT
14	OUT_DLY_EVENT_CNT
15	OUT_DLY_EVENT_CNT
16	OUT_DLY_EVENT_CNT
17	OUT_DLY_EVENT_CNT
18	OUT_DLY_EVENT_CNT
19	OUT_DLY_EVENT_CNT
20	OUT_DLY_EVENT_CNT
21	OUT_DLY_EVENT_CNT
22	OUT_DLY_EVENT_CNT
23	OUT_DLY_EVENT_CNT
24	OUT_DLY_EVENT_CNT
25	OUT_DLY_EVENT_CNT
26	OUT_DLY_EVENT_CNT
27	OUT_DLY_EVENT_CNT
28	OUT_DLY_EVENT_CNT
29	OUT_DLY_EVENT_CNT
30	OUT_DLY_EVENT_CNT
31	OUT_DLY_EVENT_CNT

OUT_DLY_ EVENT_CNT

R/W, CON83[31..0], Aon, Axion, Claxon and Cyton

When in Encoder Based Output Event Delay mode, writing an encoder count to this register creates an event in the Output Event FIFO. This new even will also include the status of the four event outputs as set by OUT_DLY_EVENT_1, OUT_DLY_EVENT_2, OUT_DLY_EVENT_3 and OUT_DLY_EVENT_4 bits.

11.11 CON84

Bit	Name
0	OUT_DLY_EN
1	Reserved
2	Reserved
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	OUT_DLY_EVENT_1
17	Reserved
18	Reserved
19	Reserved
20	OUT_DLY_EVENT_2
21	Reserved
22	Reserved
23	Reserved
24	OUT_DLY_EVENT_3
25	Reserved
26	Reserved
27	Reserved
28	OUT_DLY_EVENT_4
29	Reserved
30	Reserved
31	Reserved

OUT_DLY_EN R/W, CON84[0], Aon, Axion, Claxon and Cyton

Writing this bit to 1 will enable Encoder Based Output Event Delay mode.

OUT_DLY_EVENT_1 R/W, CON84[16], Aon, Axion, Claxon and Cyton

Writing the bit to a 1 will enable CC1 output event. The duration of the pulse will be set by the value in the register OUT_DLY_PW_1. An event with these settings will be created when an encoder count (for the location where the even should happen) is written to OUT_DLY_EVENT_CNT.

If this bit is zero when OUT_DLY_EVENT_CNT is written, no CC1 event will be created.

OUT_DLY_EVENT_2 R/W, CON84[20], Aon, Axion, Claxon and Cyton

Writing the bit to a 1 will enable a CC2 output output. The duration of the pulse will be set by the value in the register OUT_DLY_PW_2. An event with these settings will be created when an encoder count (for the location where the even should happen) is written to OUT_DLY_EVENT_CNT.

If this bit is zero when OUT_DLY_EVENT_CNT is written, no CC2 event will be created.

OUT_DLY_EVENT_3 R/W, CON84[24], Aon, Axion, Claxon and Cyton

Writing the bit to a 1 will enable a CC3 output event. The duration of the pulse will be set by the value in the register OUT_DLY_PW_3. An event with these settings will be created when an encoder count (for the location where the even should happen) is written to OUT_DLY_EVENT_CNT.

If this bit is zero when OUT_DLY_EVENT_CNT is written, no CC3 event will be created.

OUT_DLY_EVENT_4 R/W, CON84[28], Aon, Axion, Claxon and Cyton

Writing the bit to a 1 will enable a CC4 output event. The duration of the pulse will be set by the value in the register OUT_DLY_PW_4. An event with these settings will be created when an encoder count (for the location where the even should happen) is written to OUT_DLY_EVENT_CNT.

If this bit is zero when OUT_DLY_EVENT_CNT is written, no CC4 event will be created.

11.12 CON85

Bit	Name
0	OUT_DLY_WORDS_USED
1	OUT_DLY_WORDS_USED
2	OUT_DLY_WORDS_USED
3	OUT_DLY_WORDS_USED
4	OUT_DLY_WORDS_USED
5	OUT_DLY_WORDS_USED
6	OUT_DLY_WORDS_USED
7	OUT_DLY_WORDS_USED
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	OUT_DLY_FIFO_CLR
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	OUT_DLY_FIFO_OFLW

OUT_DLY_WORDS_USED RO, CON85[7..0], Aon, Axion, Claxon and Cyton

The register returns the number of entries in the Output Event FIFO

OUT_DLY_FIFO_CLR R/W, CON85[23], Aon, Axion, Claxon and Cyton

Writing the register to 1 will clear the Output Event FIFO.

OUT_DLY_FIFO_OFLW RO, CON85[31], Aon, Axion, Claxon and Cyton

If this bit returns 1, the Output Event FIFO has overflowed.

11.13 CON86

Bit	Name
0	OUT_DLY_PW_1
1	OUT_DLY_PW_1
2	OUT_DLY_PW_1
3	OUT_DLY_PW_1
4	OUT_DLY_PW_1
5	OUT_DLY_PW_1
6	OUT_DLY_PW_1
7	OUT_DLY_PW_1
8	OUT_DLY_PW_1
9	OUT_DLY_PW_1
10	OUT_DLY_PW_1
11	OUT_DLY_PW_1
12	OUT_DLY_PW_1
13	OUT_DLY_PW_1
14	OUT_DLY_PW_1
15	OUT_DLY_PW_1
16	OUT_DLY_PW_2
17	OUT_DLY_PW_2
18	OUT_DLY_PW_2
19	OUT_DLY_PW_2
20	OUT_DLY_PW_2
21	OUT_DLY_PW_2
22	OUT_DLY_PW_2
23	OUT_DLY_PW_2
24	OUT_DLY_PW_2
25	OUT_DLY_PW_2
26	OUT_DLY_PW_2
27	OUT_DLY_PW_2
28	OUT_DLY_PW_2
29	OUT_DLY_PW_2
30	OUT_DLY_PW_2
31	OUT_DLY_PW_2

OUT_DLY_PW_1 R/W, CON86[15..0], Aon, Axion, Claxon and Cyton

This is the duration of the output event plus on CC1. The units are 2 microseconds. If this register is 0, no pulse will be generated.

OUT_DLY_PW_2 R/W, CON86[31..16], Aon, Axion, Claxon and Cyton

This is the duration of the output event plus on CC2. The units are 2 microseconds. If this register is 0, no pulse will be generated.

11.14 CON87

Bit	Name
0	OUT_DLY_PW_3
1	OUT_DLY_PW_3
2	OUT_DLY_PW_3
3	OUT_DLY_PW_3
4	OUT_DLY_PW_3
5	OUT_DLY_PW_3
6	OUT_DLY_PW_3
7	OUT_DLY_PW_3
8	OUT_DLY_PW_3
9	OUT_DLY_PW_3
10	OUT_DLY_PW_3
11	OUT_DLY_PW_3
12	OUT_DLY_PW_3
13	OUT_DLY_PW_3
14	OUT_DLY_PW_3
15	OUT_DLY_PW_3
16	OUT_DLY_PW_4
17	OUT_DLY_PW_4
18	OUT_DLY_PW_4
19	OUT_DLY_PW_4
20	OUT_DLY_PW_4
21	OUT_DLY_PW_4
22	OUT_DLY_PW_4
23	OUT_DLY_PW_4
24	OUT_DLY_PW_4
25	OUT_DLY_PW_4
26	OUT_DLY_PW_4
27	OUT_DLY_PW_4
28	OUT_DLY_PW_4
29	OUT_DLY_PW_4
30	OUT_DLY_PW_4
31	OUT_DLY_PW_4

OUT_DLY_PW_3 R/W, CON87[15..0], Aon, Axion, Claxon and Cyton

This is the duration of the output event plus on CC3. The units are 2 microseconds. If this register is 0, no pulse will be generated.

OUT_DLY_PW_4 R/W, CON87[31..16], Aon, Axion, Claxon and Cyton

This is the duration of the output event plus on CC4. The units are 2 microseconds. If this register is 0, no pulse will be generated.

CXP 2.0 Registers

Chapter 12

12.1 Introduction

CoaXPress specification version 2.0 added new some new functionality. The main changes as far as new registers are concerned are the following:

Event packets - these are packets that can be sent asynchronously from the device.

Heartbeat - the device continuously sends a heartbeat packet which can be used to synchronize the device's clock with the host's clock.

The registers in the chapter are used to support this new functionality.

12.2 CON102

Bit	Name
0	0_RX_TEST_PKT_CNT
1	0_RX_TEST_PKT_CNT
2	0_RX_TEST_PKT_CNT
3	0_RX_TEST_PKT_CNT
4	0_RX_TEST_PKT_CNT
5	0_RX_TEST_PKT_CNT
6	0_RX_TEST_PKT_CNT
7	0_RX_TEST_PKT_CNT
8	0_RX_TEST_PKT_CNT
9	0_RX_TEST_PKT_CNT
10	0_RX_TEST_PKT_CNT
11	0_RX_TEST_PKT_CNT
12	0_RX_TEST_PKT_CNT
13	0_RX_TEST_PKT_CNT
14	0_RX_TEST_PKT_CNT
15	0_RX_TEST_PKT_CNT
16	0_RX_TEST_PKT_CNT
17	0_RX_TEST_PKT_CNT
18	0_RX_TEST_PKT_CNT
19	0_RX_TEST_PKT_CNT
20	0_RX_TEST_PKT_CNT
21	0_RX_TEST_PKT_CNT
22	0_RX_TEST_PKT_CNT
23	0_RX_TEST_PKT_CNT
24	0_RX_TEST_PKT_CNT
25	0_RX_TEST_PKT_CNT
26	0_RX_TEST_PKT_CNT
27	0_RX_TEST_PKT_CNT
28	0_RX_TEST_PKT_CNT
29	0_RX_TEST_PKT_CNT
30	0_RX_TEST_PKT_CNT
31	0_RX_TEST_PKT_CNT

0_RX_TEST_PKT_CNT R/W, CON102[31..0], Claxon-CXP

Number of test packets received.

12.3 CON103

Bit	Name
0	0_RX_TEST_ERR_CNT
1	0_RX_TEST_ERR_CNT
2	0_RX_TEST_ERR_CNT
3	0_RX_TEST_ERR_CNT
4	0_RX_TEST_ERR_CNT
5	0_RX_TEST_ERR_CNT
6	0_RX_TEST_ERR_CNT
7	0_RX_TEST_ERR_CNT
8	0_RX_TEST_ERR_CNT
9	0_RX_TEST_ERR_CNT
10	0_RX_TEST_ERR_CNT
11	0_RX_TEST_ERR_CNT
12	0_RX_TEST_ERR_CNT
13	0_RX_TEST_ERR_CNT
14	0_RX_TEST_ERR_CNT
15	0_RX_TEST_ERR_CNT
16	0_RX_TEST_ERR_CNT
17	0_RX_TEST_ERR_CNT
18	0_RX_TEST_ERR_CNT
19	0_RX_TEST_ERR_CNT
20	0_RX_TEST_ERR_CNT
21	0_RX_TEST_ERR_CNT
22	0_RX_TEST_ERR_CNT
23	0_RX_TEST_ERR_CNT
24	0_RX_TEST_ERR_CNT
25	0_RX_TEST_ERR_CNT
26	0_RX_TEST_ERR_CNT
27	0_RX_TEST_ERR_CNT
28	0_RX_TEST_ERR_CNT
29	0_RX_TEST_ERR_CNT
30	0_RX_TEST_ERR_CNT
31	0_RX_TEST_ERR_CNT

0_RX_TEST_ERR_CNT

R/W, CON103[31..0], Claxon-CXP

Number of 32 bits words across all test packets received that did not match the expected word.

12.4 CON112

Bit	Name
0	0_HB_DATA
1	0_HB_DATA
2	0_HB_DATA
3	0_HB_DATA
4	0_HB_DATA
5	0_HB_DATA
6	0_HB_DATA
7	0_HB_DATA
8	0_HB_DATA
9	0_HB_DATA
10	0_HB_DATA
11	0_HB_DATA
12	0_HB_DATA
13	0_HB_DATA
14	0_HB_DATA
15	0_HB_DATA
16	0_HB_DATA
17	0_HB_DATA
18	0_HB_DATA
19	0_HB_DATA
20	0_HB_DATA
21	0_HB_DATA
22	0_HB_DATA
23	0_HB_DATA
24	0_HB_DATA
25	0_HB_DATA
26	0_HB_DATA
27	0_HB_DATA
28	0_HB_DATA
29	0_HB_DATA
30	0_HB_DATA
31	0_HB_DATA

0_HB_DATA RO, CON112[31..0], Claxon-CXP

Selected heartbeat data output.

12.5 CON113

Bit	Name
0	0_HB_REG_SEL
1	0_HB_REG_SEL
2	0_HB_REG_SEL
3	Reserved
4	0_HB_CAP_EN
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

0_HB_REG_SEL R/W, CON113[2..0], Claxon-CXP

Selects which heartbeat data is visible in the HB_DATA register.

0_HB_REG_SEL	Meaning
0 (000b)	Device time from heartbeat packet, bits [31.0]
1 (001b)	Device time from heartbeat packet, bits [63..32]
2 (010b)	Connection ID from heartbeat packet
3 (011b)	This selection will cause host time to be reset
4 (100b)	Host time when heartbeat packet was received, bits [31..0]
5 (101b)	Host time when heartbeat packet was received, bits [63..32]
6 (110b)	Current host time, bits [31..0]
7 (111b)	Current host time, bits [63..32]

0_HB_CAP_EN WO, CON113[4], Claxon-CXP

Writing a 1 to this bit will cause these selected data to be captured and output to the data register.

12.6 CON119

Bit	Name
0	0_CXP_VER
1	0_CXP_VER
2	0_CXP_VER
3	0_CXP_VER
4	0_CXP_VER
5	0_CXP_VER
6	0_CXP_VER
7	0_CXP_VER
8	0_CXP_VER
9	0_CXP_VER
10	0_CXP_VER
11	0_CXP_VER
12	0_CXP_VER
13	0_CXP_VER
14	0_CXP_VER
15	0_CXP_VER
16	0_CXP_VER
17	0_CXP_VER
18	0_CXP_VER
19	0_CXP_VER
20	0_CXP_VER
21	0_CXP_VER
22	0_CXP_VER
23	0_CXP_VER
24	0_CXP_VER
25	0_CXP_VER
26	0_CXP_VER
27	0_CXP_VER
28	0_CXP_VER
29	0_CXP_VER
30	0_CXP_VER
31	0_CXP_VER

0_CXP_VER R/W, CON119[31..0], Claxon-CXP

Set's the CXP Version used to communicate with the CXP device. The table below shows the current supported versions.

0_CXP_VER	Meaning
0x00010000	CXP Version 1.0
0x00010001	CXP Version 1.1
0x00020000	CXP Version 2.0

12.7 CON124

Bit	Name
0	0_EVENT_CLR_FIFO
1	Reserved
2	Reserved
3	0_EVENT_CLR_RCV_CNT
4	0_EVENT_RCV_CNT
5	0_EVENT_RCV_CNT
6	0_EVENT_RCV_CNT
7	0_EVENT_RCV_CNT
8	0_EVENT_RCV_CNT
9	0_EVENT_RCV_CNT
10	0_EVENT_RCV_CNT
11	0_EVENT_RCV_CNT
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	0_EVENT_WORD_CNT
17	0_EVENT_WORD_CNT
18	0_EVENT_WORD_CNT
19	0_EVENT_WORD_CNT
20	0_EVENT_WORD_CNT
21	0_EVENT_WORD_CNT
22	0_EVENT_WORD_CNT
23	0_EVENT_WORD_CNT
24	0_EVENT_WORD_CNT
25	0_EVENT_WORD_CNT
26	0_EVENT_WORD_CNT
27	0_EVENT_WORD_CNT
28	0_EVENT_WORD_CNT
29	0_EVENT_WORD_CNT
30	0_EVENT_WORD_CNT
31	0_EVENT_WORD_CNT

0_EVENT_CLR_FIFO WO, CON124[0], Claxon-CXP

Writing a 1 to this bit will clear the event FIFO.

0_EVENT_CLR_RCV_CNT WO, CON124[3], Claxon-CXP

Writing a 1 to this bit will clear the received event count.

0_EVENT_RCV_CNT RO, CON124[11..4], Claxon-CXP

Number of event packets received.

0_EVENT_WORD_CNT RO, CON124[31..16], Claxon-CXP

Number of words event FIFO.

12.8 CON125

Bit	Name
0	0_EVENT_DATA
1	0_EVENT_DATA
2	0_EVENT_DATA
3	0_EVENT_DATA
4	0_EVENT_DATA
5	0_EVENT_DATA
6	0_EVENT_DATA
7	0_EVENT_DATA
8	0_EVENT_DATA
9	0_EVENT_DATA
10	0_EVENT_DATA
11	0_EVENT_DATA
12	0_EVENT_DATA
13	0_EVENT_DATA
14	0_EVENT_DATA
15	0_EVENT_DATA
16	0_EVENT_DATA
17	0_EVENT_DATA
18	0_EVENT_DATA
19	0_EVENT_DATA
20	0_EVENT_DATA
21	0_EVENT_DATA
22	0_EVENT_DATA
23	0_EVENT_DATA
24	0_EVENT_DATA
25	0_EVENT_DATA
26	0_EVENT_DATA
27	0_EVENT_DATA
28	0_EVENT_DATA
29	0_EVENT_DATA
30	0_EVENT_DATA
31	0_EVENT_DATA

0_EVENT_DATA RO, CON125[31..0], Claxon-CXP

Returns the first word in the event FIFO.

12.9 CON126

Bit	Name
0	0_TRIG_RQST_CNT
1	0_TRIG_RQST_CNT
2	0_TRIG_RQST_CNT
3	0_TRIG_RQST_CNT
4	0_TRIG_RQST_CNT
5	0_TRIG_RQST_CNT
6	0_TRIG_RQST_CNT
7	0_TRIG_RQST_CNT
8	0_TRIG_SENT_CNT
9	0_TRIG_SENT_CNT
10	0_TRIG_SENT_CNT
11	0_TRIG_SENT_CNT
12	0_TRIG_SENT_CNT
13	0_TRIG_SENT_CNT
14	0_TRIG_SENT_CNT
15	0_TRIG_SENT_CNT
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	0_TRIG_AUTO_EN
21	Reserved
22	Reserved
23	Reserved
24	0_TRIG_CLR_CNTS
25	Reserved
26	Reserved
27	Reserved
28	0_TRIG_SEND_ONE_ACK
29	Reserved
30	Reserved
31	Reserved

0_TRIG_RQST_CNT RO, CON126[7..0], Claxon-CXP

Number of requested trigger Acks.

0_TRIG_SENT_CNT RO, CON126[15..8], Claxon-CXP

Number of actual trigger Acks sent.

0_TRIG_AUTO_EN R/W, CON126[20], Claxon-CXP

Set to 1 to enable auto trigger Ack.

0_TRIG_CLR_CNTS WO, CON126[24], Claxon-CXP

Write a 1 to this register will clear TRIG_RQST_CNT and TRIG_SENT_CNT.

0_TRIG_SEND_ONE_ACK WO, CON126[28], Claxon-CXP

Writing a 1 to this bit will send a single trigger Ack when TRIG_AUTO_EN = 0.

12.10 CON130

Bit	Name
0	0_EVENT_RQST_CNT
1	0_EVENT_RQST_CNT
2	0_EVENT_RQST_CNT
3	0_EVENT_RQST_CNT
4	0_EVENT_RQST_CNT
5	0_EVENT_RQST_CNT
6	0_EVENT_RQST_CNT
7	0_EVENT_RQST_CNT
8	0_EVENT_SENT_CNT
9	0_EVENT_SENT_CNT
10	0_EVENT_SENT_CNT
11	0_EVENT_SENT_CNT
12	0_EVENT_SENT_CNT
13	0_EVENT_SENT_CNT
14	0_EVENT_SENT_CNT
15	0_EVENT_SENT_CNT
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	0_EVENT_AUTO_EN
21	Reserved
22	Reserved
23	Reserved
24	0_EVENT_CLR_CNTS
25	Reserved
26	Reserved
27	Reserved
28	0_EVENT_SEND_ONE_ACK
29	Reserved
30	Reserved
31	Reserved

0_EVENT_RQST_CNT RO, CON130[7..0], Claxon-CXP

Number of requested event Acks.

0_EVENT_SENT_CNT RO, CON130[15..8], Claxon-CXP

Number of actual event Acks sent.

0_EVENT_AUTO_EN R/W, CON130[20], Claxon-CXP

Enables auto event Ack.

0_EVENT_CLR_CNTS WO, CON130[24], Claxon-CXP

Writing a 1 to this register will clear EVENT_RQST_CNT and EVENT_SENT_CNT.

0_EVENT_SEND_ONE_ACK WO, CON130[28], Claxon-CXP

Set to send a single event Ack when EVENT_AUTO_EN = 0.

12.11 CON131

Bit	Name
0	0_EVENT_TAG_TO_SEND
1	0_EVENT_TAG_TO_SEND
2	0_EVENT_TAG_TO_SEND
3	0_EVENT_TAG_TO_SEND
4	0_EVENT_TAG_TO_SEND
5	0_EVENT_TAG_TO_SEND
6	0_EVENT_TAG_TO_SEND
7	0_EVENT_TAG_TO_SEND
8	0_EVENT_TAG_RX_LAST
9	0_EVENT_TAG_RX_LAST
10	0_EVENT_TAG_RX_LAST
11	0_EVENT_TAG_RX_LAST
12	0_EVENT_TAG_RX_LAST
13	0_EVENT_TAG_RX_LAST
14	0_EVENT_TAG_RX_LAST
15	0_EVENT_TAG_RX_LAST
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

0_EVENT_TAG_TO_SEND R/W, CON131[7..0], Claxon-CXP

Event Tag to use on next event Ack when sending single event Ack (EVENT_AUTO_EN = 0)

0_EVENT_TAG_RX_LAST RO, CON131[15..8], Claxon-CXP

Tag of last event packet received.

12.12 CON134

Bit	Name
0	1_RX_TEST_PKT_CNT
1	1_RX_TEST_PKT_CNT
2	1_RX_TEST_PKT_CNT
3	1_RX_TEST_PKT_CNT
4	1_RX_TEST_PKT_CNT
5	1_RX_TEST_PKT_CNT
6	1_RX_TEST_PKT_CNT
7	1_RX_TEST_PKT_CNT
8	1_RX_TEST_PKT_CNT
9	1_RX_TEST_PKT_CNT
10	1_RX_TEST_PKT_CNT
11	1_RX_TEST_PKT_CNT
12	1_RX_TEST_PKT_CNT
13	1_RX_TEST_PKT_CNT
14	1_RX_TEST_PKT_CNT
15	1_RX_TEST_PKT_CNT
16	1_RX_TEST_PKT_CNT
17	1_RX_TEST_PKT_CNT
18	1_RX_TEST_PKT_CNT
19	1_RX_TEST_PKT_CNT
20	1_RX_TEST_PKT_CNT
21	1_RX_TEST_PKT_CNT
22	1_RX_TEST_PKT_CNT
23	1_RX_TEST_PKT_CNT
24	1_RX_TEST_PKT_CNT
25	1_RX_TEST_PKT_CNT
26	1_RX_TEST_PKT_CNT
27	1_RX_TEST_PKT_CNT
28	1_RX_TEST_PKT_CNT
29	1_RX_TEST_PKT_CNT
30	1_RX_TEST_PKT_CNT
31	1_RX_TEST_PKT_CNT

1_RX_TEST_PKT_CNT R/W, CON134[31..0], Claxon-CXP

Number of test packets received.

12.13 CON135

Bit	Name
0	1_RX_TEST_ERR_CNT
1	1_RX_TEST_ERR_CNT
2	1_RX_TEST_ERR_CNT
3	1_RX_TEST_ERR_CNT
4	1_RX_TEST_ERR_CNT
5	1_RX_TEST_ERR_CNT
6	1_RX_TEST_ERR_CNT
7	1_RX_TEST_ERR_CNT
8	1_RX_TEST_ERR_CNT
9	1_RX_TEST_ERR_CNT
10	1_RX_TEST_ERR_CNT
11	1_RX_TEST_ERR_CNT
12	1_RX_TEST_ERR_CNT
13	1_RX_TEST_ERR_CNT
14	1_RX_TEST_ERR_CNT
15	1_RX_TEST_ERR_CNT
16	1_RX_TEST_ERR_CNT
17	1_RX_TEST_ERR_CNT
18	1_RX_TEST_ERR_CNT
19	1_RX_TEST_ERR_CNT
20	1_RX_TEST_ERR_CNT
21	1_RX_TEST_ERR_CNT
22	1_RX_TEST_ERR_CNT
23	1_RX_TEST_ERR_CNT
24	1_RX_TEST_ERR_CNT
25	1_RX_TEST_ERR_CNT
26	1_RX_TEST_ERR_CNT
27	1_RX_TEST_ERR_CNT
28	1_RX_TEST_ERR_CNT
29	1_RX_TEST_ERR_CNT
30	1_RX_TEST_ERR_CNT
31	1_RX_TEST_ERR_CNT

1_RX_TEST_ERR_CNT

R/W, CON135[31..0], Claxon-CXP

Number of 32 bits words across all test packets received that did not match the expected word.

12.14 CON144

Bit	Name
0	1_HB_DATA
1	1_HB_DATA
2	1_HB_DATA
3	1_HB_DATA
4	1_HB_DATA
5	1_HB_DATA
6	1_HB_DATA
7	1_HB_DATA
8	1_HB_DATA
9	1_HB_DATA
10	1_HB_DATA
11	1_HB_DATA
12	1_HB_DATA
13	1_HB_DATA
14	1_HB_DATA
15	1_HB_DATA
16	1_HB_DATA
17	1_HB_DATA
18	1_HB_DATA
19	1_HB_DATA
20	1_HB_DATA
21	1_HB_DATA
22	1_HB_DATA
23	1_HB_DATA
24	1_HB_DATA
25	1_HB_DATA
26	1_HB_DATA
27	1_HB_DATA
28	1_HB_DATA
29	1_HB_DATA
30	1_HB_DATA
31	1_HB_DATA

1_HB_DATA RO, CON144[31..0], Claxon-CXP

Selected hearbeat data output.

12.15 CON145

Bit	Name
0	1_HB_REG_SEL
1	1_HB_REG_SEL
2	1_HB_REG_SEL
3	Reserved
4	1_HB_CAP_EN
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

1_HB_REG_SEL R/W, CON145[2..0], Claxon-CXP

Selects which heartbeat data is visible in the HB_DATA register.

1_HB_REG_SEL	Meaning
0 (000b)	Device time from heartbeat packet, bits [31.0]
1 (001b)	Device time from heartbeat packet, bits [63..32]
2 (010b)	Connection ID from heartbeat packet
3 (011b)	This selection will cause host time to be reset
4 (100b)	Host time when heartbeat packet was received, bits [31..0]
5 (101b)	Host time when heartbeat packet was received, bits [63..32]
6 (110b)	Current host time, bits [31..0]
7 (111b)	Current host time, bits [63..32]

1_HB_CAP_EN WO, CON145[4], Claxon-CXP

Writing a 1 to this bit will cause these selected data to be captured and output to the data register.

12.16 CON151

Bit	Name
0	1_CXP_VER
1	1_CXP_VER
2	1_CXP_VER
3	1_CXP_VER
4	1_CXP_VER
5	1_CXP_VER
6	1_CXP_VER
7	1_CXP_VER
8	1_CXP_VER
9	1_CXP_VER
10	1_CXP_VER
11	1_CXP_VER
12	1_CXP_VER
13	1_CXP_VER
14	1_CXP_VER
15	1_CXP_VER
16	1_CXP_VER
17	1_CXP_VER
18	1_CXP_VER
19	1_CXP_VER
20	1_CXP_VER
21	1_CXP_VER
22	1_CXP_VER
23	1_CXP_VER
24	1_CXP_VER
25	1_CXP_VER
26	1_CXP_VER
27	1_CXP_VER
28	1_CXP_VER
29	1_CXP_VER
30	1_CXP_VER
31	1_CXP_VER

1_CXP_VER R/W, CON151[31..0], Claxon-CXP

Set's the CXP Version used to communicate with the CXP device.

1_CXP_VER	Meaning
0x00010000	CXP Version 1.0
0x00010001	CXP Version 1.1
0x00020000	CXP Version 2.0

12.17 CON156

Bit	Name
0	1_EVENT_CLR_FIFO
1	Reserved
2	Reserved
3	1_EVENT_CLR_RCV_CNT
4	1_EVENT_RCV_CNT
5	1_EVENT_RCV_CNT
6	1_EVENT_RCV_CNT
7	1_EVENT_RCV_CNT
8	1_EVENT_RCV_CNT
9	1_EVENT_RCV_CNT
10	1_EVENT_RCV_CNT
11	1_EVENT_RCV_CNT
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	1_EVENT_WORD_CNT
17	1_EVENT_WORD_CNT
18	1_EVENT_WORD_CNT
19	1_EVENT_WORD_CNT
20	1_EVENT_WORD_CNT
21	1_EVENT_WORD_CNT
22	1_EVENT_WORD_CNT
23	1_EVENT_WORD_CNT
24	1_EVENT_WORD_CNT
25	1_EVENT_WORD_CNT
26	1_EVENT_WORD_CNT
27	1_EVENT_WORD_CNT
28	1_EVENT_WORD_CNT
29	1_EVENT_WORD_CNT
30	1_EVENT_WORD_CNT
31	1_EVENT_WORD_CNT

1_EVENT_CLR_FIFO WO, CON156[0], Claxon-CXP

Writing a 1 to this bit will clear the event FIFO.

1_EVENT_CLR_RCV_CNT WO, CON156[3], Claxon-CXP

Writing a 1 to this bit will clear the received event count (number of words in event FIFO)..

1_EVENT_RCV_CNT RO, CON156[11..4], Claxon-CXP

Number of event packets received.

1_EVENT_WORD_CNT RO, CON156[31..16], Claxon-CXP

Number of words event FIFO.

12.18 CON157

Bit	Name
0	1_EVENT_DATA
1	1_EVENT_DATA
2	1_EVENT_DATA
3	1_EVENT_DATA
4	1_EVENT_DATA
5	1_EVENT_DATA
6	1_EVENT_DATA
7	1_EVENT_DATA
8	1_EVENT_DATA
9	1_EVENT_DATA
10	1_EVENT_DATA
11	1_EVENT_DATA
12	1_EVENT_DATA
13	1_EVENT_DATA
14	1_EVENT_DATA
15	1_EVENT_DATA
16	1_EVENT_DATA
17	1_EVENT_DATA
18	1_EVENT_DATA
19	1_EVENT_DATA
20	1_EVENT_DATA
21	1_EVENT_DATA
22	1_EVENT_DATA
23	1_EVENT_DATA
24	1_EVENT_DATA
25	1_EVENT_DATA
26	1_EVENT_DATA
27	1_EVENT_DATA
28	1_EVENT_DATA
29	1_EVENT_DATA
30	1_EVENT_DATA
31	1_EVENT_DATA

1_EVENT_DATA RO, CON157[31..0], Claxon-CXP

Returns the first word in the event FIFO.

12.19 CON158

Bit	Name
0	1_TRIG_RQST_CNT
1	1_TRIG_RQST_CNT
2	1_TRIG_RQST_CNT
3	1_TRIG_RQST_CNT
4	1_TRIG_RQST_CNT
5	1_TRIG_RQST_CNT
6	1_TRIG_RQST_CNT
7	1_TRIG_RQST_CNT
8	1_TRIG_SENT_CNT
9	1_TRIG_SENT_CNT
10	1_TRIG_SENT_CNT
11	1_TRIG_SENT_CNT
12	1_TRIG_SENT_CNT
13	1_TRIG_SENT_CNT
14	1_TRIG_SENT_CNT
15	1_TRIG_SENT_CNT
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	1_TRIG_AUTO_EN
21	Reserved
22	Reserved
23	Reserved
24	1_TRIG_CLR_CNTS
25	Reserved
26	Reserved
27	Reserved
28	1_TRIG_SEND_ONE_ACK
29	Reserved
30	Reserved
31	Reserved

1_TRIG_RQST_CNT RO, CON158[7..0], Claxon-CXP

Number of requested trigger Acks.

1_TRIG_SENT_CNT RO, CON158[15..8], Claxon-CXP

Number of actual trigger Acks sent.

1_TRIG_AUTO_EN R/W, CON158[20], Claxon-CXP

Set to 1 to enable auto trigger Ack.

1_TRIG_CLR_CNTS WO, CON158[24], Claxon-CXP

Write a 1 to this register will clear TRIG_RQST_CNT and TRIG_SENT_CNT.

1_TRIG_SEND_ONE_ACK WO, CON158[28], Claxon-CXP

Writing a 1 to this bit will send a single trigger Ack when TRIG_AUTO_EN = 0.

12.20 CON162

Bit	Name
0	1_EVENT_RQST_CNT
1	1_EVENT_RQST_CNT
2	1_EVENT_RQST_CNT
3	1_EVENT_RQST_CNT
4	1_EVENT_RQST_CNT
5	1_EVENT_RQST_CNT
6	1_EVENT_RQST_CNT
7	1_EVENT_RQST_CNT
8	1_EVENT_SENT_CNT
9	1_EVENT_SENT_CNT
10	1_EVENT_SENT_CNT
11	1_EVENT_SENT_CNT
12	1_EVENT_SENT_CNT
13	1_EVENT_SENT_CNT
14	1_EVENT_SENT_CNT
15	1_EVENT_SENT_CNT
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	1_EVENT_AUTO_EN
21	Reserved
22	Reserved
23	Reserved
24	1_EVENT_CLR_CNTS
25	Reserved
26	Reserved
27	Reserved
28	1_EVENT_SEND_ONE_ACK
29	Reserved
30	Reserved
31	Reserved

1_EVENT_RQST_CNT RO, CON162[7..0], Claxon-CXP

Number of requested event Acks.

1_EVENT_SENT_CNT RO, CON162[15..8], Claxon-CXP

Number of actual event Acks sent.

1_EVENT_AUTO_EN R/W, CON162[20], Claxon-CXP

Enables auto event Ack.

1_EVENT_CLR_CNTS WO, CON162[24], Claxon-CXP

Writing a 1 to this register will clear EVENT_RQST_CNT and EVENT_SENT_CNT.

1_EVENT_SEND_ONE_ACK WO, CON162[28], Claxon-CXP

Set to send a single event Ack when EVENT_AUTO_EN = 0.

12.21 CON163

Bit	Name
0	1_EVENT_TAG_TO_SEND
1	1_EVENT_TAG_TO_SEND
2	1_EVENT_TAG_TO_SEND
3	1_EVENT_TAG_TO_SEND
4	1_EVENT_TAG_TO_SEND
5	1_EVENT_TAG_TO_SEND
6	1_EVENT_TAG_TO_SEND
7	1_EVENT_TAG_TO_SEND
8	1_EVENT_TAG_RX_LAST
9	1_EVENT_TAG_RX_LAST
10	1_EVENT_TAG_RX_LAST
11	1_EVENT_TAG_RX_LAST
12	1_EVENT_TAG_RX_LAST
13	1_EVENT_TAG_RX_LAST
14	1_EVENT_TAG_RX_LAST
15	1_EVENT_TAG_RX_LAST
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

1_EVENT_TAG_TO_SEND R/W, CON163[7..0], Claxon-CXP

Event Tag to use on next event Ack when sending single event Ack (EVENT_AUTO_EN = 0)

1_EVENT_TAG_RX_LAST RO, CON163[15..8], Claxon-CXP

Tag of last event packet received.

12.22 CON166

Bit	Name
0	2_RX_TEST_PKT_CNT
1	2_RX_TEST_PKT_CNT
2	2_RX_TEST_PKT_CNT
3	2_RX_TEST_PKT_CNT
4	2_RX_TEST_PKT_CNT
5	2_RX_TEST_PKT_CNT
6	2_RX_TEST_PKT_CNT
7	2_RX_TEST_PKT_CNT
8	2_RX_TEST_PKT_CNT
9	2_RX_TEST_PKT_CNT
10	2_RX_TEST_PKT_CNT
11	2_RX_TEST_PKT_CNT
12	2_RX_TEST_PKT_CNT
13	2_RX_TEST_PKT_CNT
14	2_RX_TEST_PKT_CNT
15	2_RX_TEST_PKT_CNT
16	2_RX_TEST_PKT_CNT
17	2_RX_TEST_PKT_CNT
18	2_RX_TEST_PKT_CNT
19	2_RX_TEST_PKT_CNT
20	2_RX_TEST_PKT_CNT
21	2_RX_TEST_PKT_CNT
22	2_RX_TEST_PKT_CNT
23	2_RX_TEST_PKT_CNT
24	2_RX_TEST_PKT_CNT
25	2_RX_TEST_PKT_CNT
26	2_RX_TEST_PKT_CNT
27	2_RX_TEST_PKT_CNT
28	2_RX_TEST_PKT_CNT
29	2_RX_TEST_PKT_CNT
30	2_RX_TEST_PKT_CNT
31	2_RX_TEST_PKT_CNT

2_RX_TEST_PKT_CNT R/W, CON166[31..0], Claxon-CXP

Number of test packets received.

12.23 CON167

Bit	Name
0	2_RX_TEST_ERR_CNT
1	2_RX_TEST_ERR_CNT
2	2_RX_TEST_ERR_CNT
3	2_RX_TEST_ERR_CNT
4	2_RX_TEST_ERR_CNT
5	2_RX_TEST_ERR_CNT
6	2_RX_TEST_ERR_CNT
7	2_RX_TEST_ERR_CNT
8	2_RX_TEST_ERR_CNT
9	2_RX_TEST_ERR_CNT
10	2_RX_TEST_ERR_CNT
11	2_RX_TEST_ERR_CNT
12	2_RX_TEST_ERR_CNT
13	2_RX_TEST_ERR_CNT
14	2_RX_TEST_ERR_CNT
15	2_RX_TEST_ERR_CNT
16	2_RX_TEST_ERR_CNT
17	2_RX_TEST_ERR_CNT
18	2_RX_TEST_ERR_CNT
19	2_RX_TEST_ERR_CNT
20	2_RX_TEST_ERR_CNT
21	2_RX_TEST_ERR_CNT
22	2_RX_TEST_ERR_CNT
23	2_RX_TEST_ERR_CNT
24	2_RX_TEST_ERR_CNT
25	2_RX_TEST_ERR_CNT
26	2_RX_TEST_ERR_CNT
27	2_RX_TEST_ERR_CNT
28	2_RX_TEST_ERR_CNT
29	2_RX_TEST_ERR_CNT
30	2_RX_TEST_ERR_CNT
31	2_RX_TEST_ERR_CNT

2_RX_TEST_ERR_CNT

R/W, CON167[31..0], Claxon-CXP

Number of 32 bits words across all test packets received that did not match the expected word.

12.24 CON176

Bit	Name
0	2_HB_DATA
1	2_HB_DATA
2	2_HB_DATA
3	2_HB_DATA
4	2_HB_DATA
5	2_HB_DATA
6	2_HB_DATA
7	2_HB_DATA
8	2_HB_DATA
9	2_HB_DATA
10	2_HB_DATA
11	2_HB_DATA
12	2_HB_DATA
13	2_HB_DATA
14	2_HB_DATA
15	2_HB_DATA
16	2_HB_DATA
17	2_HB_DATA
18	2_HB_DATA
19	2_HB_DATA
20	2_HB_DATA
21	2_HB_DATA
22	2_HB_DATA
23	2_HB_DATA
24	2_HB_DATA
25	2_HB_DATA
26	2_HB_DATA
27	2_HB_DATA
28	2_HB_DATA
29	2_HB_DATA
30	2_HB_DATA
31	2_HB_DATA

2_HB_DATA RO, CON176[31..0], Claxon-CXP

Selected heartbeat data output.

12.25 CON177

Bit	Name
0	2_HB_REG_SEL
1	2_HB_REG_SEL
2	2_HB_REG_SEL
3	Reserved
4	2_HB_CAP_EN
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

2_HB_REG_SEL R/W, CON177[2..0], Claxon-CXP

Selects which heartbeat data is visible in the HB_DATA register.

2_HB_REG_SEL	Meaning
0 (000b)	Device time from heartbeat packet, bits [31.0]
1 (001b)	Device time from heartbeat packet, bits [63..32]
2 (010b)	Connection ID from heartbeat packet
3 (011b)	This selection will cause host time to be reset
4 (100b)	Host time when heartbeat packet was received, bits [31..0]
5 (101b)	Host time when heartbeat packet was received, bits [63..32]
6 (110b)	Current host time, bits [31..0]
7 (111b)	Current host time, bits [63..32]

2_HB_CAP_EN WO, CON177[4], Claxon-CXP

Writing a 1 to this bit will cause these selected data to be captured and output to the data register.

12.26 CON183

Bit	Name
0	2_CXP_VER
1	2_CXP_VER
2	2_CXP_VER
3	2_CXP_VER
4	2_CXP_VER
5	2_CXP_VER
6	2_CXP_VER
7	2_CXP_VER
8	2_CXP_VER
9	2_CXP_VER
10	2_CXP_VER
11	2_CXP_VER
12	2_CXP_VER
13	2_CXP_VER
14	2_CXP_VER
15	2_CXP_VER
16	2_CXP_VER
17	2_CXP_VER
18	2_CXP_VER
19	2_CXP_VER
20	2_CXP_VER
21	2_CXP_VER
22	2_CXP_VER
23	2_CXP_VER
24	2_CXP_VER
25	2_CXP_VER
26	2_CXP_VER
27	2_CXP_VER
28	2_CXP_VER
29	2_CXP_VER
30	2_CXP_VER
31	2_CXP_VER

2_CXP_VER R/W, CON183[31..0], Claxon-CXP

Set's the CXP Version used to communicate with the CXP device.

2_CXP_VER	Meaning
0x00010000	CXP Version 1.0
0x00010001	CXP Version 1.1
0x00020000	CXP Version 2.0

12.27 CON188

Bit	Name
0	2_EVENT_CLR_FIFO
1	Reserved
2	Reserved
3	2_EVENT_CLR_RCV_CNT
4	2_EVENT_RCV_CNT
5	2_EVENT_RCV_CNT
6	2_EVENT_RCV_CNT
7	2_EVENT_RCV_CNT
8	2_EVENT_RCV_CNT
9	2_EVENT_RCV_CNT
10	2_EVENT_RCV_CNT
11	2_EVENT_RCV_CNT
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	2_EVENT_WORD_CNT
17	2_EVENT_WORD_CNT
18	2_EVENT_WORD_CNT
19	2_EVENT_WORD_CNT
20	2_EVENT_WORD_CNT
21	2_EVENT_WORD_CNT
22	2_EVENT_WORD_CNT
23	2_EVENT_WORD_CNT
24	2_EVENT_WORD_CNT
25	2_EVENT_WORD_CNT
26	2_EVENT_WORD_CNT
27	2_EVENT_WORD_CNT
28	2_EVENT_WORD_CNT
29	2_EVENT_WORD_CNT
30	2_EVENT_WORD_CNT
31	2_EVENT_WORD_CNT

2_EVENT_CLR_FIFO WO, CON188[0], Claxon-CXP

Writing a 1 to this bit will clear the event FIFO.

2_EVENT_CLR_RCV_CNT WO, CON188[3], Claxon-CXP

Writing a 1 to this bit will clear the received event count.

2_EVENT_RCV_CNT RO, CON188[11..4], Claxon-CXP

Number of event packets received.

2_EVENT_WORD_CNT RO, CON188[31..16], Claxon-CXP

Number of words event FIFO.

12.28 CON189

Bit	Name
0	2_EVENT_DATA
1	2_EVENT_DATA
2	2_EVENT_DATA
3	2_EVENT_DATA
4	2_EVENT_DATA
5	2_EVENT_DATA
6	2_EVENT_DATA
7	2_EVENT_DATA
8	2_EVENT_DATA
9	2_EVENT_DATA
10	2_EVENT_DATA
11	2_EVENT_DATA
12	2_EVENT_DATA
13	2_EVENT_DATA
14	2_EVENT_DATA
15	2_EVENT_DATA
16	2_EVENT_DATA
17	2_EVENT_DATA
18	2_EVENT_DATA
19	2_EVENT_DATA
20	2_EVENT_DATA
21	2_EVENT_DATA
22	2_EVENT_DATA
23	2_EVENT_DATA
24	2_EVENT_DATA
25	2_EVENT_DATA
26	2_EVENT_DATA
27	2_EVENT_DATA
28	2_EVENT_DATA
29	2_EVENT_DATA
30	2_EVENT_DATA
31	2_EVENT_DATA

2_EVENT_DATA RO, CON189[31..0], Claxon-CXP

Returns the first word in the event FIFO.

12.29 CON190

Bit	Name
0	2_TRIG_RQST_CNT
1	2_TRIG_RQST_CNT
2	2_TRIG_RQST_CNT
3	2_TRIG_RQST_CNT
4	2_TRIG_RQST_CNT
5	2_TRIG_RQST_CNT
6	2_TRIG_RQST_CNT
7	2_TRIG_RQST_CNT
8	2_TRIG_SENT_CNT
9	2_TRIG_SENT_CNT
10	2_TRIG_SENT_CNT
11	2_TRIG_SENT_CNT
12	2_TRIG_SENT_CNT
13	2_TRIG_SENT_CNT
14	2_TRIG_SENT_CNT
15	2_TRIG_SENT_CNT
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	2_TRIG_AUTO_EN
21	Reserved
22	Reserved
23	Reserved
24	2_TRIG_CLR_CNTS
25	Reserved
26	Reserved
27	Reserved
28	2_TRIG_SEND_ONE_ACK
29	Reserved
30	Reserved
31	Reserved

2_TRIG_RQST_CNT RO, CON190[7..0], Claxon-CXP

Number of requested trigger Acks.

2_TRIG_SENT_CNT RO, CON190[15..8], Claxon-CXP

Number of actual trigger Acks sent.

2_TRIG_AUTO_EN R/W, CON190[20], Claxon-CXP

Set to 1 to enable auto trigger Ack.

2_TRIG_CLR_CNTS WO, CON190[24], Claxon-CXP

Write a 1 to this register will clear TRIG_RQST_CNT and TRIG_SENT_CNT.

2_TRIG_SEND_ONE_ACK WO, CON190[28], Claxon-CXP

Writing a 1 to this bit will send a single trigger Ack when TRIG_AUTO_EN = 0.

12.30 CON194

Bit	Name
0	2_EVENT_RQST_CNT
1	2_EVENT_RQST_CNT
2	2_EVENT_RQST_CNT
3	2_EVENT_RQST_CNT
4	2_EVENT_RQST_CNT
5	2_EVENT_RQST_CNT
6	2_EVENT_RQST_CNT
7	2_EVENT_RQST_CNT
8	2_EVENT_SENT_CNT
9	2_EVENT_SENT_CNT
10	2_EVENT_SENT_CNT
11	2_EVENT_SENT_CNT
12	2_EVENT_SENT_CNT
13	2_EVENT_SENT_CNT
14	2_EVENT_SENT_CNT
15	2_EVENT_SENT_CNT
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	2_EVENT_AUTO_EN
21	Reserved
22	Reserved
23	Reserved
24	2_EVENT_CLR_CNTS
25	Reserved
26	Reserved
27	Reserved
28	2_EVENT_SEND_ONE_ACK
29	Reserved
30	Reserved
31	Reserved

2_EVENT_RQST_CNT RO, CON194[7..0], Claxon-CXP

Number of requested event Acks..

2_EVENT_SENT_CNT RO, CON194[15..8], Claxon-CXP

Number of actual event Acks sent.

2_EVENT_AUTO_EN R/W, CON194[20], Claxon-CXP

Enables auto event Ack.

2_EVENT_CLR_CNTS WO, CON194[24], Claxon-CXP

Writing a 1 to this register will clear EVENT_RQST_CNT and EVENT_SENT_CNT.

2_EVENT_SEND_ONE_ACK WO, CON194[28], Claxon-CXP

Set to send a single event Ack when EVENT_AUTO_EN = 0.

12.31 CON195

Bit	Name
0	2_EVENT_TAG_TO_SEND
1	2_EVENT_TAG_TO_SEND
2	2_EVENT_TAG_TO_SEND
3	2_EVENT_TAG_TO_SEND
4	2_EVENT_TAG_TO_SEND
5	2_EVENT_TAG_TO_SEND
6	2_EVENT_TAG_TO_SEND
7	2_EVENT_TAG_TO_SEND
8	2_EVENT_TAG_RX_LAST
9	2_EVENT_TAG_RX_LAST
10	2_EVENT_TAG_RX_LAST
11	2_EVENT_TAG_RX_LAST
12	2_EVENT_TAG_RX_LAST
13	2_EVENT_TAG_RX_LAST
14	2_EVENT_TAG_RX_LAST
15	2_EVENT_TAG_RX_LAST
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

2_EVENT_TAG_TO_SEND R/W, CON195[7..0], Claxon-CXP

Event Tag to use on next event Ack when sending single event Ack (EVENT_AUTO_EN = 0)

2_EVENT_TAG_RX_LAST RO, CON195[15..8], Claxon-CXP

Tag of last event packet received.

12.32 CON198

Bit	Name
0	3_RX_TEST_PKT_CNT
1	3_RX_TEST_PKT_CNT
2	3_RX_TEST_PKT_CNT
3	3_RX_TEST_PKT_CNT
4	3_RX_TEST_PKT_CNT
5	3_RX_TEST_PKT_CNT
6	3_RX_TEST_PKT_CNT
7	3_RX_TEST_PKT_CNT
8	3_RX_TEST_PKT_CNT
9	3_RX_TEST_PKT_CNT
10	3_RX_TEST_PKT_CNT
11	3_RX_TEST_PKT_CNT
12	3_RX_TEST_PKT_CNT
13	3_RX_TEST_PKT_CNT
14	3_RX_TEST_PKT_CNT
15	3_RX_TEST_PKT_CNT
16	3_RX_TEST_PKT_CNT
17	3_RX_TEST_PKT_CNT
18	3_RX_TEST_PKT_CNT
19	3_RX_TEST_PKT_CNT
20	3_RX_TEST_PKT_CNT
21	3_RX_TEST_PKT_CNT
22	3_RX_TEST_PKT_CNT
23	3_RX_TEST_PKT_CNT
24	3_RX_TEST_PKT_CNT
25	3_RX_TEST_PKT_CNT
26	3_RX_TEST_PKT_CNT
27	3_RX_TEST_PKT_CNT
28	3_RX_TEST_PKT_CNT
29	3_RX_TEST_PKT_CNT
30	3_RX_TEST_PKT_CNT
31	3_RX_TEST_PKT_CNT

3_RX_TEST_PKT_CNT R/W, CON198[31..0], Claxon-CXP

Number of test packets received.

12.33 CON199

Bit	Name
0	3_RX_TEST_ERR_CNT
1	3_RX_TEST_ERR_CNT
2	3_RX_TEST_ERR_CNT
3	3_RX_TEST_ERR_CNT
4	3_RX_TEST_ERR_CNT
5	3_RX_TEST_ERR_CNT
6	3_RX_TEST_ERR_CNT
7	3_RX_TEST_ERR_CNT
8	3_RX_TEST_ERR_CNT
9	3_RX_TEST_ERR_CNT
10	3_RX_TEST_ERR_CNT
11	3_RX_TEST_ERR_CNT
12	3_RX_TEST_ERR_CNT
13	3_RX_TEST_ERR_CNT
14	3_RX_TEST_ERR_CNT
15	3_RX_TEST_ERR_CNT
16	3_RX_TEST_ERR_CNT
17	3_RX_TEST_ERR_CNT
18	3_RX_TEST_ERR_CNT
19	3_RX_TEST_ERR_CNT
20	3_RX_TEST_ERR_CNT
21	3_RX_TEST_ERR_CNT
22	3_RX_TEST_ERR_CNT
23	3_RX_TEST_ERR_CNT
24	3_RX_TEST_ERR_CNT
25	3_RX_TEST_ERR_CNT
26	3_RX_TEST_ERR_CNT
27	3_RX_TEST_ERR_CNT
28	3_RX_TEST_ERR_CNT
29	3_RX_TEST_ERR_CNT
30	3_RX_TEST_ERR_CNT
31	3_RX_TEST_ERR_CNT

3_RX_TEST_ERR_CNT

R/W, CON199[31..0], Claxon-CXP

Number of 32 bits words across all test packets received that did not match the expected word.

12.34 CON208

Bit	Name
0	3_HB_DATA
1	3_HB_DATA
2	3_HB_DATA
3	3_HB_DATA
4	3_HB_DATA
5	3_HB_DATA
6	3_HB_DATA
7	3_HB_DATA
8	3_HB_DATA
9	3_HB_DATA
10	3_HB_DATA
11	3_HB_DATA
12	3_HB_DATA
13	3_HB_DATA
14	3_HB_DATA
15	3_HB_DATA
16	3_HB_DATA
17	3_HB_DATA
18	3_HB_DATA
19	3_HB_DATA
20	3_HB_DATA
21	3_HB_DATA
22	3_HB_DATA
23	3_HB_DATA
24	3_HB_DATA
25	3_HB_DATA
26	3_HB_DATA
27	3_HB_DATA
28	3_HB_DATA
29	3_HB_DATA
30	3_HB_DATA
31	3_HB_DATA

3_HB_DATA RO, CON208[31..0], Claxon-CXP

Selected heartbeat data output.

12.35 CON209

Bit	Name
0	3_HB_REG_SEL
1	3_HB_REG_SEL
2	3_HB_REG_SEL
3	Reserved
4	3_HB_CAP_EN
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

3_HB_REG_SEL R/W, CON209[2..0], Claxon-CXP

Selects which heartbeat data is visible in the HB_DATA register.

3_HB_REG_SEL	Meaning
0 (000b)	Device time from heartbeat packet, bits [31.0]
1 (001b)	Device time from heartbeat packet, bits [63..32]
2 (010b)	Connection ID from heartbeat packet
3 (011b)	This selection will cause host time to be reset
4 (100b)	Host time when heartbeat packet was received, bits [31..0]
5 (101b)	Host time when heartbeat packet was received, bits [63..32]
6 (110b)	Current host time, bits [31..0]
7 (111b)	Current host time, bits [63..32]

3_HB_CAP_EN WO, CON209[4], Claxon-CXP

Writing a 1 to this bit will cause these selected data to be captured and output to the data register.

12.36 CON215

Bit	Name
0	3_CXP_VER
1	3_CXP_VER
2	3_CXP_VER
3	3_CXP_VER
4	3_CXP_VER
5	3_CXP_VER
6	3_CXP_VER
7	3_CXP_VER
8	3_CXP_VER
9	3_CXP_VER
10	3_CXP_VER
11	3_CXP_VER
12	3_CXP_VER
13	3_CXP_VER
14	3_CXP_VER
15	3_CXP_VER
16	3_CXP_VER
17	3_CXP_VER
18	3_CXP_VER
19	3_CXP_VER
20	3_CXP_VER
21	3_CXP_VER
22	3_CXP_VER
23	3_CXP_VER
24	3_CXP_VER
25	3_CXP_VER
26	3_CXP_VER
27	3_CXP_VER
28	3_CXP_VER
29	3_CXP_VER
30	3_CXP_VER
31	3_CXP_VER

3_CXP_VER R/W, CON215[31..0], Claxon-CXP

Set's the CXP Version used to communicate with the CXP device.

3_CXP_VER	Meaning
0x00010000	CXP Version 1.0
0x00010001	CXP Version 1.1
0x00020000	CXP Version 2.0

12.37 CON220

Bit	Name
0	3_EVENT_CLR_FIFO
1	Reserved
2	Reserved
3	3_EVENT_CLR_RCV_CNT
4	3_EVENT_RCV_CNT
5	3_EVENT_RCV_CNT
6	3_EVENT_RCV_CNT
7	3_EVENT_RCV_CNT
8	3_EVENT_RCV_CNT
9	3_EVENT_RCV_CNT
10	3_EVENT_RCV_CNT
11	3_EVENT_RCV_CNT
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	3_EVENT_WORD_CNT
17	3_EVENT_WORD_CNT
18	3_EVENT_WORD_CNT
19	3_EVENT_WORD_CNT
20	3_EVENT_WORD_CNT
21	3_EVENT_WORD_CNT
22	3_EVENT_WORD_CNT
23	3_EVENT_WORD_CNT
24	3_EVENT_WORD_CNT
25	3_EVENT_WORD_CNT
26	3_EVENT_WORD_CNT
27	3_EVENT_WORD_CNT
28	3_EVENT_WORD_CNT
29	3_EVENT_WORD_CNT
30	3_EVENT_WORD_CNT
31	3_EVENT_WORD_CNT

3_EVENT_CLR_FIFO WO, CON220[0], Claxon-CXP

Writing a 1 to this bit will clear the event FIFO.

3_EVENT_CLR_RCV_CNT WO, CON220[3], Claxon-CXP

Writing a 1 to this bit will clear the received event count (number of words in event FIFO).

3_EVENT_RCV_CNT RO, CON220[11..4], Claxon-CXP

Number of event packets received.

3_EVENT_WORD_CNT RO, CON220[31..16], Claxon-CXP

Number of words event FIFO.

12.38 CON221

Bit	Name
0	3_EVENT_DATA
1	3_EVENT_DATA
2	3_EVENT_DATA
3	3_EVENT_DATA
4	3_EVENT_DATA
5	3_EVENT_DATA
6	3_EVENT_DATA
7	3_EVENT_DATA
8	3_EVENT_DATA
9	3_EVENT_DATA
10	3_EVENT_DATA
11	3_EVENT_DATA
12	3_EVENT_DATA
13	3_EVENT_DATA
14	3_EVENT_DATA
15	3_EVENT_DATA
16	3_EVENT_DATA
17	3_EVENT_DATA
18	3_EVENT_DATA
19	3_EVENT_DATA
20	3_EVENT_DATA
21	3_EVENT_DATA
22	3_EVENT_DATA
23	3_EVENT_DATA
24	3_EVENT_DATA
25	3_EVENT_DATA
26	3_EVENT_DATA
27	3_EVENT_DATA
28	3_EVENT_DATA
29	3_EVENT_DATA
30	3_EVENT_DATA
31	3_EVENT_DATA

3_EVENT_DATA RO, CON221[31..0], Claxon-CXP

Returns the first word in the event FIFO.

12.39 CON222

Bit	Name
0	3_TRIG_RQST_CNT
1	3_TRIG_RQST_CNT
2	3_TRIG_RQST_CNT
3	3_TRIG_RQST_CNT
4	3_TRIG_RQST_CNT
5	3_TRIG_RQST_CNT
6	3_TRIG_RQST_CNT
7	3_TRIG_RQST_CNT
8	3_TRIG_SENT_CNT
9	3_TRIG_SENT_CNT
10	3_TRIG_SENT_CNT
11	3_TRIG_SENT_CNT
12	3_TRIG_SENT_CNT
13	3_TRIG_SENT_CNT
14	3_TRIG_SENT_CNT
15	3_TRIG_SENT_CNT
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	3_TRIG_AUTO_EN
21	Reserved
22	Reserved
23	Reserved
24	3_TRIG_CLR_CNTS
25	Reserved
26	Reserved
27	Reserved
28	3_TRIG_SEND_ONE_ACK
29	Reserved
30	Reserved
31	Reserved

3_TRIG_RQST_CNT RO, CON222[7..0], Claxon-CXP

Number of requested trigger Acks.

3_TRIG_SENT_CNT RO, CON222[15..8], Claxon-CXP

Number of actual trigger Acks sent.

3_TRIG_AUTO_EN R/W, CON222[20], Claxon-CXP

Set to 1 to enable auto trigger Ack.

3_TRIG_CLR_CNTS WO, CON222[24], Claxon-CXP

Write a 1 to this register will clear TRIG_RQST_CNT and TRIG_SENT_CNT.

3_TRIG_SEND_ONE_ACK WO, CON222[28], Claxon-CXP

Writing a 1 to this bit will send a single trigger Ack when TRIG_AUTO_EN = 0.

12.40 CON226

Bit	Name
0	3_EVENT_RQST_CNT
1	3_EVENT_RQST_CNT
2	3_EVENT_RQST_CNT
3	3_EVENT_RQST_CNT
4	3_EVENT_RQST_CNT
5	3_EVENT_RQST_CNT
6	3_EVENT_RQST_CNT
7	3_EVENT_RQST_CNT
8	3_EVENT_SENT_CNT
9	3_EVENT_SENT_CNT
10	3_EVENT_SENT_CNT
11	3_EVENT_SENT_CNT
12	3_EVENT_SENT_CNT
13	3_EVENT_SENT_CNT
14	3_EVENT_SENT_CNT
15	3_EVENT_SENT_CNT
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	3_EVENT_AUTO_EN
21	Reserved
22	Reserved
23	Reserved
24	3_EVENT_CLR_CNTS
25	Reserved
26	Reserved
27	Reserved
28	3_EVENT_SEND_ONE_ACK
29	Reserved
30	Reserved
31	Reserved

3_EVENT_RQST_CNT RO, CON226[7..0], Claxon-CXP

Number of requested event Acks.

3_EVENT_SENT_CNT RO, CON226[15..8], Claxon-CXP

Number of actual event Acks sent.

3_EVENT_AUTO_EN R/W, CON226[20], Claxon-CXP

Enables auto event Ack.

3_EVENT_CLR_CNTS WO, CON226[24], Claxon-CXP

Writing a 1 to this register will clear EVENT_RQST_CNT and EVENT_SENT_CNT.

3_EVENT_SEND_ONE_ACK WO, CON226[28], Claxon-CXP

Set to send a single event Ack when EVENT_AUTO_EN = 0.

12.41 CON227

Bit	Name
0	3_EVENT_TAG_TO_SEND
1	3_EVENT_TAG_TO_SEND
2	3_EVENT_TAG_TO_SEND
3	3_EVENT_TAG_TO_SEND
4	3_EVENT_TAG_TO_SEND
5	3_EVENT_TAG_TO_SEND
6	3_EVENT_TAG_TO_SEND
7	3_EVENT_TAG_TO_SEND
8	3_EVENT_TAG_RX_LAST
9	3_EVENT_TAG_RX_LAST
10	3_EVENT_TAG_RX_LAST
11	3_EVENT_TAG_RX_LAST
12	3_EVENT_TAG_RX_LAST
13	3_EVENT_TAG_RX_LAST
14	3_EVENT_TAG_RX_LAST
15	3_EVENT_TAG_RX_LAST
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

3_EVENT_TAG_TO_SEND R/W, CON227[7..0], Claxon-CXP

Event Tag to use on next event Ack when sending single event Ack (EVENT_AUTO_EN = 0)

3_EVENT_TAG_RX_LAST RO, CON227[15..8], Claxon-CXP

Tag of last event packet received.

12.42 TMSTMP_FCTL_BASE

Bit	Name
0	TMSTMP_WORDS_AVAIL
1	TMSTMP_WORDS_AVAIL
2	TMSTMP_WORDS_AVAIL
3	TMSTMP_WORDS_AVAIL
4	TMSTMP_WORDS_AVAIL
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	TMSTMP_POP_FIFO
16	Reserved
17	Reserved
18	Reserved
19	TMSTMP_PRESENT
20	Reserved
21	Reserved
22	Reserved
23	TMSTMP_FIFO_CLR
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	TMSTMP_FIFO_OFLW

TMSTMP_WORDS_AVAIL R/W, TMSTMP_FCTL_BASE[4..0], Claxon-CXP

Number of start of frame timestamps in the FIFO.

TMSTMP_POP_FIFO R/W, TMSTMP_FCTL_BASE[15], Claxon-CXP

Write a 1 to this bit to pop one timestamp off the FIFO

TMSTMP_PRESENT RO, TMSTMP_FCTL_BASE[19], Claxon-CXP

If this bit is 1, the current board firmware contains the hardware timestamp circuit.

TMSTMP_FIFO_CLR R/W, TMSTMP_FCTL_BASE[23], Claxon-CXP

Clears the start of frame timestamp FIFO.

TMSTMP_FIFO_OFLW RO, TMSTMP_FCTL_BASE[31], Claxon-CXP

If this bit is 1, the start of frame timestamp FIFO has overflowed.

12.43 TMSTMP_LO_BASE

Bit	Name
0	TMSTMP_LO
1	TMSTMP_LO
2	TMSTMP_LO
3	TMSTMP_LO
4	TMSTMP_LO
5	TMSTMP_LO
6	TMSTMP_LO
7	TMSTMP_LO
8	TMSTMP_LO
9	TMSTMP_LO
10	TMSTMP_LO
11	TMSTMP_LO
12	TMSTMP_LO
13	TMSTMP_LO
14	TMSTMP_LO
15	TMSTMP_LO
16	TMSTMP_LO
17	TMSTMP_LO
18	TMSTMP_LO
19	TMSTMP_LO
20	TMSTMP_LO
21	TMSTMP_LO
22	TMSTMP_LO
23	TMSTMP_LO
24	TMSTMP_LO
25	TMSTMP_LO
26	TMSTMP_LO
27	TMSTMP_LO
28	TMSTMP_LO
29	TMSTMP_LO
30	TMSTMP_LO
31	TMSTMP_LO

TMSTMP_LO R/W, TMSTMP_LO_BASE[31..0], Claxon-CXP

Start of frame timestamp low word.

12.44 TMSTMP_HI_BASE

Bit	Name
0	TMSTMP_HI
1	TMSTMP_HI
2	TMSTMP_HI
3	TMSTMP_HI
4	TMSTMP_HI
5	TMSTMP_HI
6	TMSTMP_HI
7	TMSTMP_HI
8	TMSTMP_HI
9	TMSTMP_HI
10	TMSTMP_HI
11	TMSTMP_HI
12	TMSTMP_HI
13	TMSTMP_HI
14	TMSTMP_HI
15	TMSTMP_HI
16	TMSTMP_HI
17	TMSTMP_HI
18	TMSTMP_HI
19	TMSTMP_HI
20	TMSTMP_HI
21	TMSTMP_HI
22	TMSTMP_HI
23	TMSTMP_HI
24	TMSTMP_HI
25	TMSTMP_HI
26	TMSTMP_HI
27	TMSTMP_HI
28	TMSTMP_HI
29	TMSTMP_HI
30	TMSTMP_HI
31	TMSTMP_HI

TMSTMP_HI R/W, TMSTMP_HI_BASE[31..0], Claxon-CXP

Start of frame timestamp high word.

Specifications

Chapter 13

13.1 Introduction

This chapter describes the general specifications of the Aon, Claxon and Cyton families. The numerical values for the Aon specifications are listed in Table 13-1, the Claxon specifications are in Table 13-2 and the Cyton in Table 13-3. If more information is available for a given specification there will be an entry in the column marked "Details".

Table 13-1 Aon-CXP Specifications

Specifications	Value	Units	Details
PCIe Compatibility, slot type	x4, x8 and x16	Slot size	Section 13.2
PCIe Compatibility, generation	Gen2 and 3		Section 13.2
Maximum Input CXP Rate	6.5	Gb/S	
Minimum Input CXP Rate	1.25	Gb/S	
Maximum Pixels Per Line (1 tap)	268,435,456	Pixels (8-bit)	
Maximum Lines Per Frame	16,777,216	Lines	
Minimum trigger pulse	10	Nanoseconds	
Minimum encoder pulse	10	Nanoseconds	
Aon-CXP Current (3.3 Volt)	0.56	Amps	Section 13.3
Aon-CXP Current (12 Volt)	0.22	Amps	Section 13.3
Temperature range	0 to 50	Degrees Celsius	
Temperature range storage	-20 to 100	Degrees Celsius	
Humidity	25% to 80%		
Mechanical dimensions	5.5 x 3	Inches	
Mechanical dimensions	14 x 7.7	Centimeters	
Maximum CXP Power @24 Volts	13	Watts/Link	
LVDS Drivers	SN65LVDS31D		
LVDS Receivers	SNLVDS3486		
TTL Drivers	SN74LVTH241		
TTL Receivers	SN74LVTH241		

Table 13-2 Claxon-CXP Specifications

Specifications	Value	Units	Details
PCIe Compatibility, slot type	x1, x2, x4 x8 and x16	Slot size	Section 13.2
PCIe Compatibility, generation	Gen2 and 3		Section 13.2
Maximum Input CXP Rate	12.5	Gb/S	
Minimum Input CXP Rate	1.25	Gb/S	
Maximum Pixels Per Line (1 tap)	268,435,456	Pixels (8-bit)	
Maximum Lines Per Frame	16,777,216	Lines	
Minimum trigger pulse	10	Nanoseconds	
Minimum encoder pulse	10	Nanoseconds	
Claxon-CXP4 Current (3.3 Volt)	0.55	Amps	Section 13.3
Claxon-CXP4 Current (12 Volt)	1.4	Amps	Section 13.3
Claxon-CXP1 Current (3.3 Volt)	0.54	Amps	Section 13.3
Claxon-CXP1 Current (12 Volt)	0.47	Amps	Section 13.3
Claxon-FXP4 Current (3.3 Volt)	0.55	Amps	
Claxon-FXP4 Current (12 Volt)	1.4	Amps	
Temperature range	0 to 50	Degrees Celsius	
Temperature range storage	-20 to 100	Degrees Celsius	
Humidity	25% to 80%		
Mechanical dimensions (CXP1)	5.5 x 3	Inches	
Mechanical dimensions (CXP1)	14 x 7.7	Centimeters	
Mechanical dimensions (CXP2, CXP4)	6.8 x 4.2	Inches	
Mechanical dimensions (CXP2, CXP4)	17.4 x 10.6	Centimeters	
Maximum CXP Power @24 Volts	13	Watts/Link	
LVDS Drivers	SN65LVDS31D		
LVDS Receivers	SNLVDS3486		
TTL Drivers	SN74LVTH241		
TTL Receivers	SN74LVTH241		
Claxon-Fiber Connector	40G QFSP+		Section 13.4

Table 13-3 Cyton-CXP Specifications

Specifications	Value	Units	Details
PCIe Compatibility, slot type	x8 and x16	Slot size	Section 13.2
PCIe Compatibility, generation	Gen2 and 3		Section 13.2
Maximum Input CXP Rate	6.25	Gb/S	
Minimum Input CXP Rate	1.25	Gb/S	
Maximum Pixels Per Line (1 tap)	268,435,456	Pixels (8-bit)	
Maximum Lines Per Frame	16,777,216	Lines	
Minimum trigger pulse	10	Nanoseconds	
Minimum encoder pulse	10	Nanoseconds	
Cyton-CXP4 Current (3.3 Volt)	0.55	Amps	Section 13.3
Cyton-CXP4 Current (12 Volt)	1.25	Amps	Section 13.3
Cyton-CXP2 Current (3.3 Volt)	0.32	Amps	Section 13.3
Cyton-CXP2 Current (12 Volt)	0.88	Amps	Section 13.3
Temperature range	0 to 50	Degrees Celsius	
Temperature range storage	-20 to 100	Degrees Celsius	
Humidity	25% to 80%		
Mechanical dimensions	6.8 x 4.2	Inches	
Mechanical dimensions	17.4 x 10.6	Centimeters	
Maximum CXP Power @24 Volts	13	Watts/Link	
LVDS Drivers	SN65LVDS31D		
LVDS Receivers	SNLVDS3486		
TTL Drivers	SN74LVTH241		
TTL Receivers	SN74LVTH241		

13.2 PCI Express Compatibility

Table 13-4 lists the PCIe compatibility of each CXP model.

Table 13-4 PCIe Compatibility

	Compatible		Compatible with Possible Reduced Performance		Not Compatible	
Model	**Gen2**	**Gen3**	**Gen2**	**Gen3**	**Gen2**	**Gen3**
AON-PC2-CXP1	x4, x8, x16	x4, x8, x16	x1	x1		
CLX-PC2-CXP1	x4, x8, x16	x4, x8, x16	x1	x1		
CLX-PC3-CXP2		x8, x16	x1, x4, x8, x16	x1, x4		
CXL-PC3-CXP4		x8, x16	x1, x4, x8, x16	x1, x4		
CXL-PC3-FXP4		x8, x16	x1, x4, x8, x16	x1, x4		
CTN-PC2-CXP2 Rev 5. x or earlier	x8, x16	x8, x16			x1, x4	x1, x4
CTN-PC2-CXP4 Rev 5. x or earlier	x8, x16	x8, x16			x1, x4	x1, x4
CTN-PC2-CXP2 Rev 6.0 or later	x8, x16	x8, x16	x1, x4	x1, x4		
CTN-PC2-CXP4 Rev 6.0 or later	x8, x16	x8, x16	x1, x4	x1, x4		

13.3 Power Requirements

The power requirements listed in Table 13-1, Table 13-2 and Table 13-3 are the requirements of the board's circuitry only. In addition, the Aon, Claxon and Cyton can provide up to 13 watts of power to each CXP link. In the case of the CXP4 models, the 12 Volt rail of the PCIe bus cannot provide enough power if all four links are drawing maximum power. The board has an auxiliary connector which can be use to provide additional PoCXP power for these situations (P4). There is a jumper which is used to switch the power source from the PCIe bus to the auxiliary connector. See Section 14.8 for more information on this jumper.

Note: If the total amount of all cameras connected to the four link boards exceeds 15 Watts, then the auxiliary power connector must be used. For example, if four single link cameras each taking 4 watts are connected, then auxiliary power should be used. Similarly, if a single camera quad link camera that takes 20 watts on two links is connected, then auxiliary power must be used.

The one and two link boards can derive enough power from the PCIe bus to power any CXP single link camera. However an auxiliary power connector is still provided (P3) for customers that would prefer a separate camera power source. In order to use power from this connector there is a jumper which must be changed, see Section 14.6 for more information.

13.4 Claxon-Fiber Connector Compatibility

The Claxon-Fiber use a QFSP+ cage for connections to cameras. The following list enumerates the compatible cable types.

- 40G QSFP+ AOC (any length)
- 40G QSFP+ Transceivers (any length cable is supported, but care must be taken to use the correct fiber type)

Mechanical

Chapter 14

14.1 Introduction

This chapter describes the mechanical characteristics of the Aon, Claxon and Cyton. This includes description of all of the connectors on the boar and pin-outs for these connectors.

Table 14-1 lists the mechanical layout figures for each board.

Table 14-1 List of Layouts

Model	Figure
Aon-CXP1	Figure 14-1
Claxon-CXP1	Figure 14-2
Claxon-CXP2	Figure 14-3
Claxon-CXP4	Figure 14-4
Claxon-FXP4 (Claxon-Fiber)	Figure 14-5
Cyton-CXP2 (Rev 5.x or earlier)	Figure 14-6
Cyton-CXP4 (Rev 5.x or earlier)	Figure 14-7
Cyton-CXP2 (Rev 6.0 or later)	Figure 14-8
Cyton-CXP4 (Rev 6.0 or later)	Figure 14-9

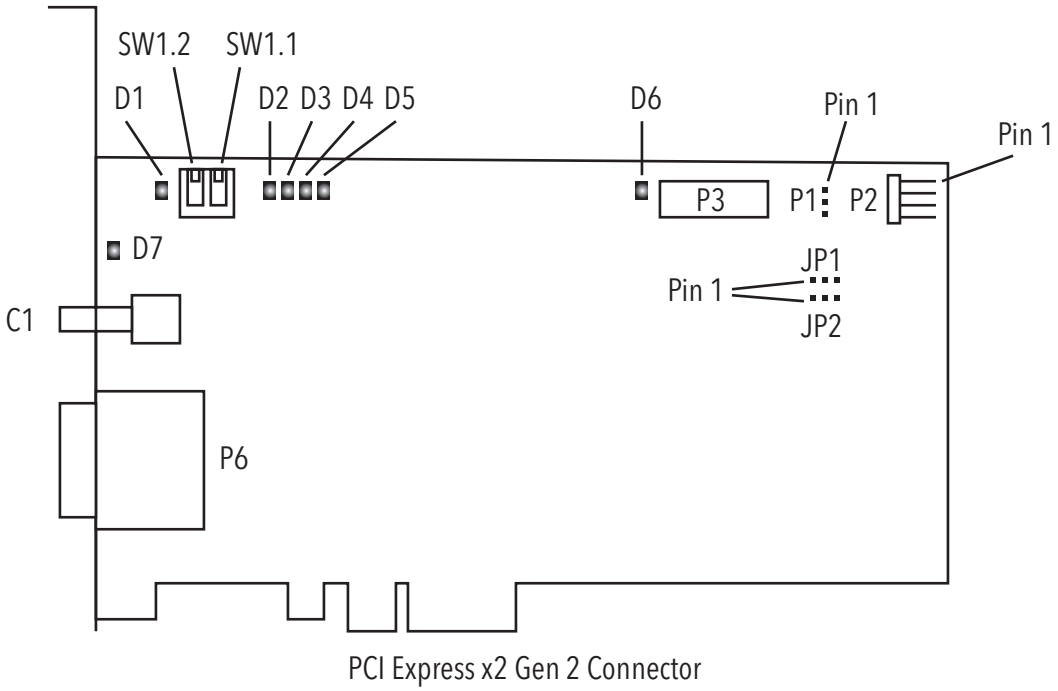

Figure 14-1 Aon-CXP Board Layout

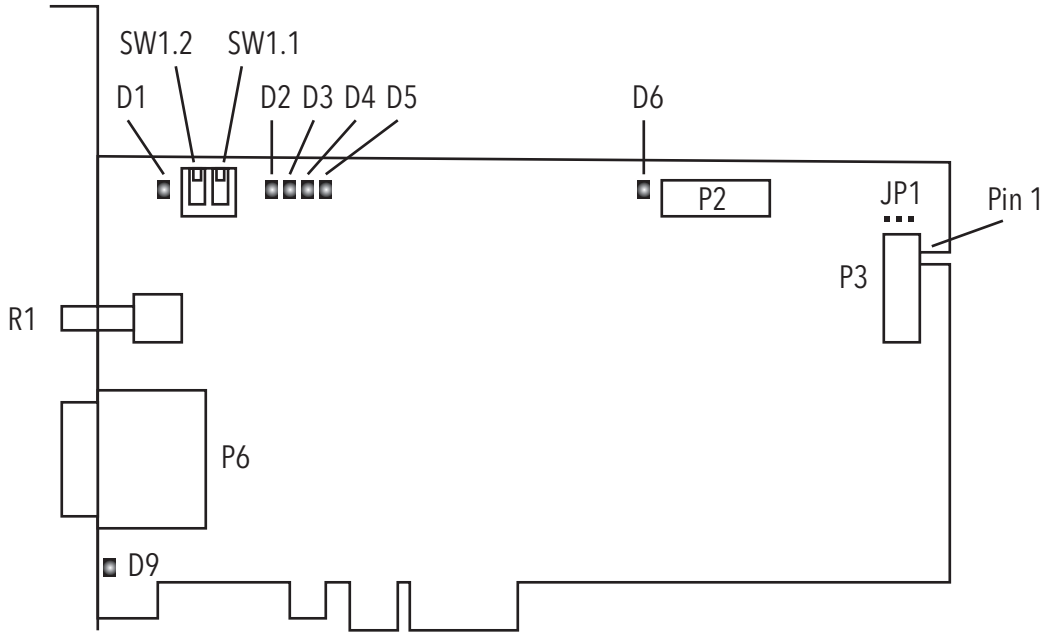

Figure 14-2 Claxon-CXP1 Board Layout

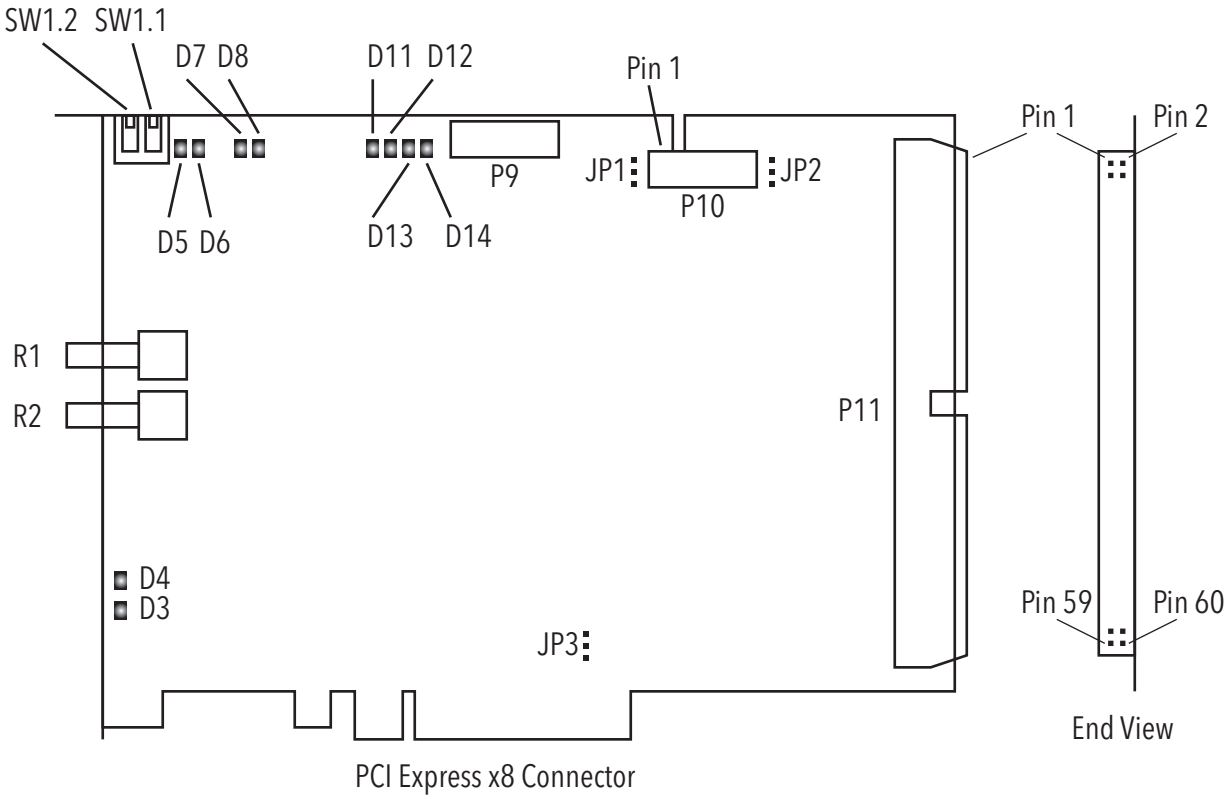

Figure 14-3 Claxon-CXP2 Board Layout

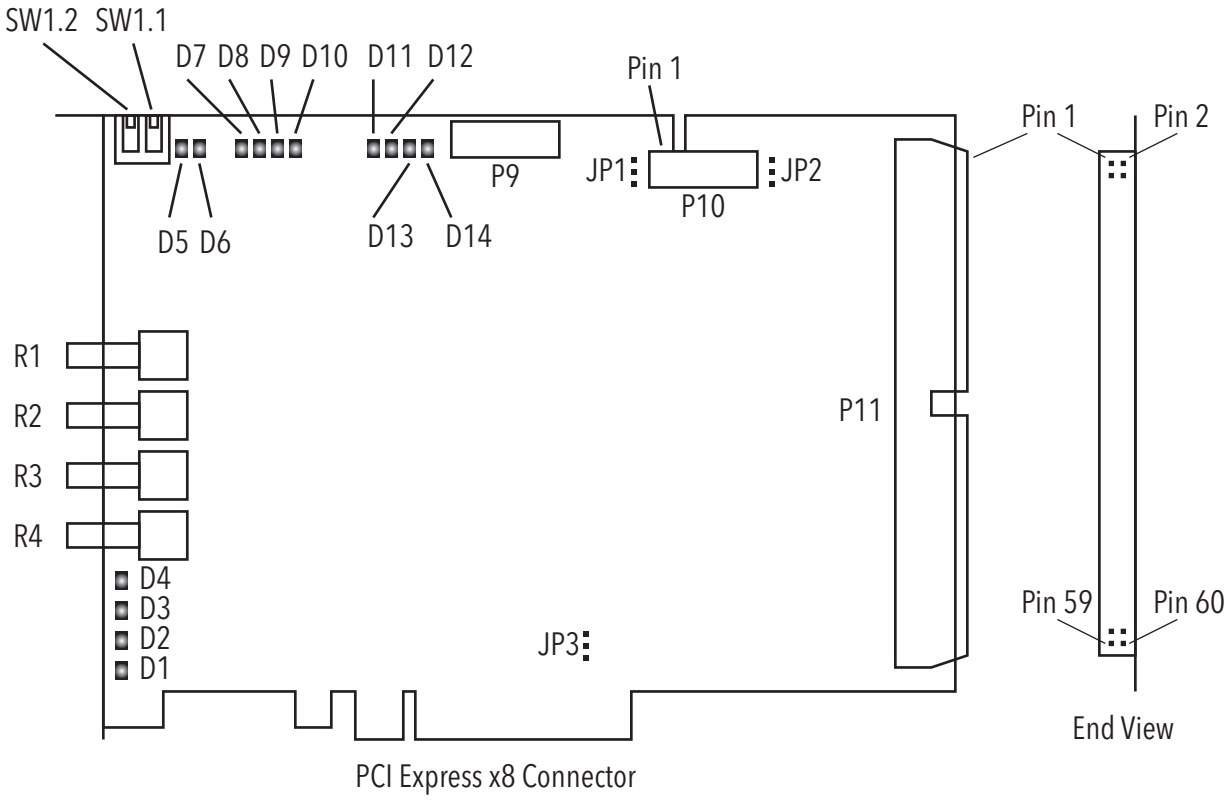

Figure 14-4 Claxon-CXP4 Board Layout

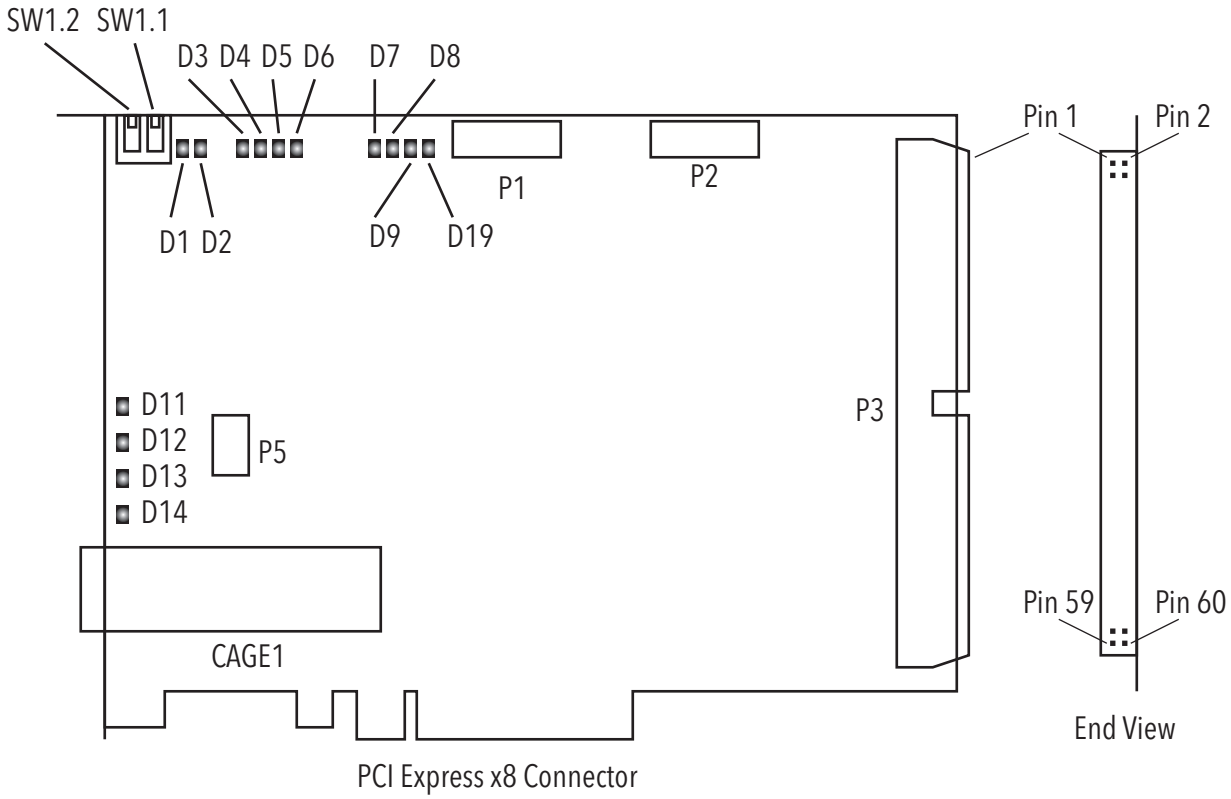

Figure 14-5 Claxon-FXP4 Board Layout

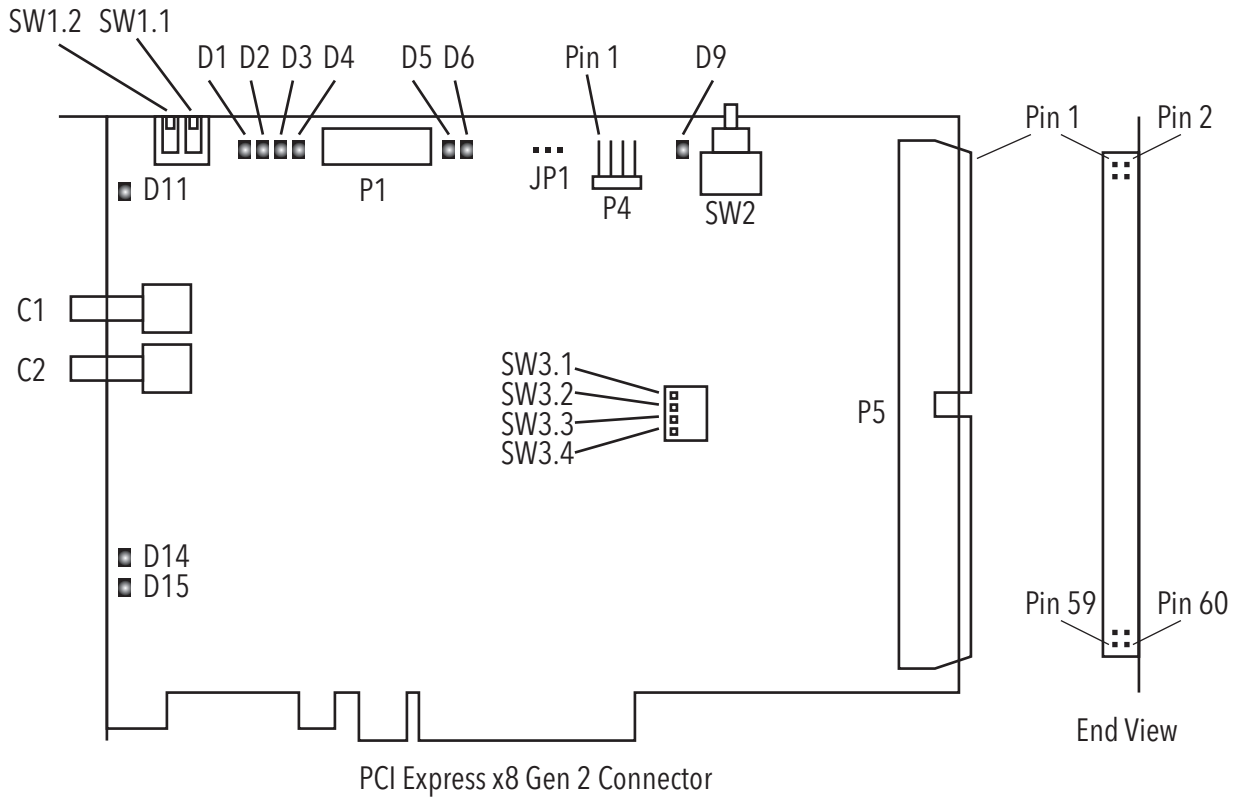

Figure 14-6 Cyton-CXP2 Board Layout (Rev 5.x or earlier)

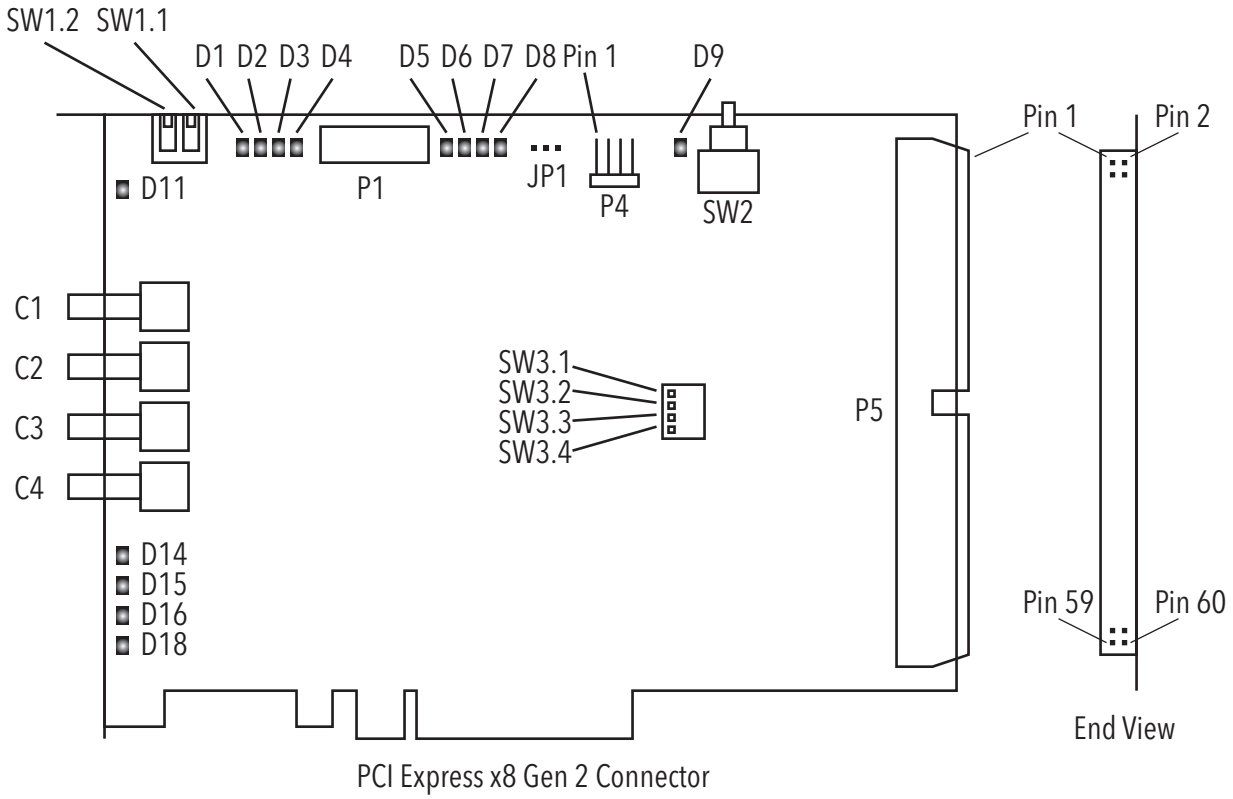

Figure 14-7 Cyton-CXP4 Board Layout (Rev 5.x or earlier)

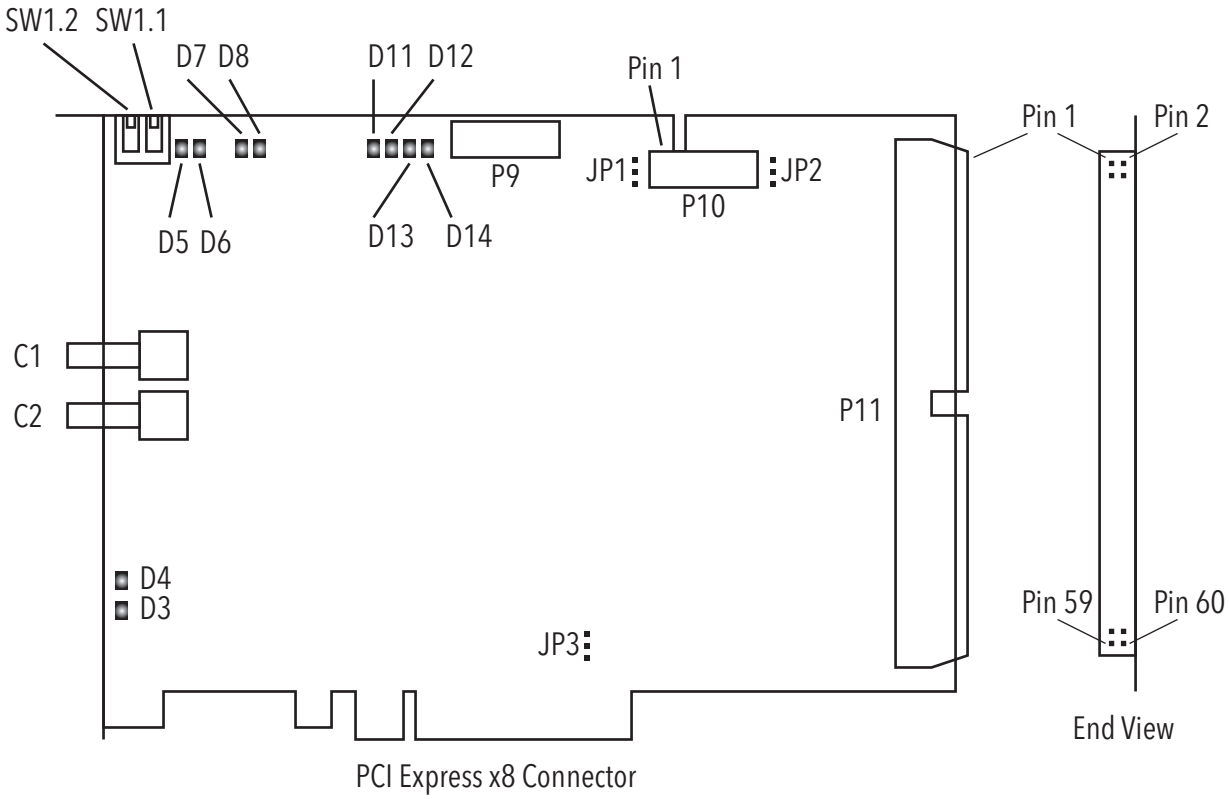

Figure 14-8 Cyton-CXP2 Board Layout (Rev 6.0 or later)

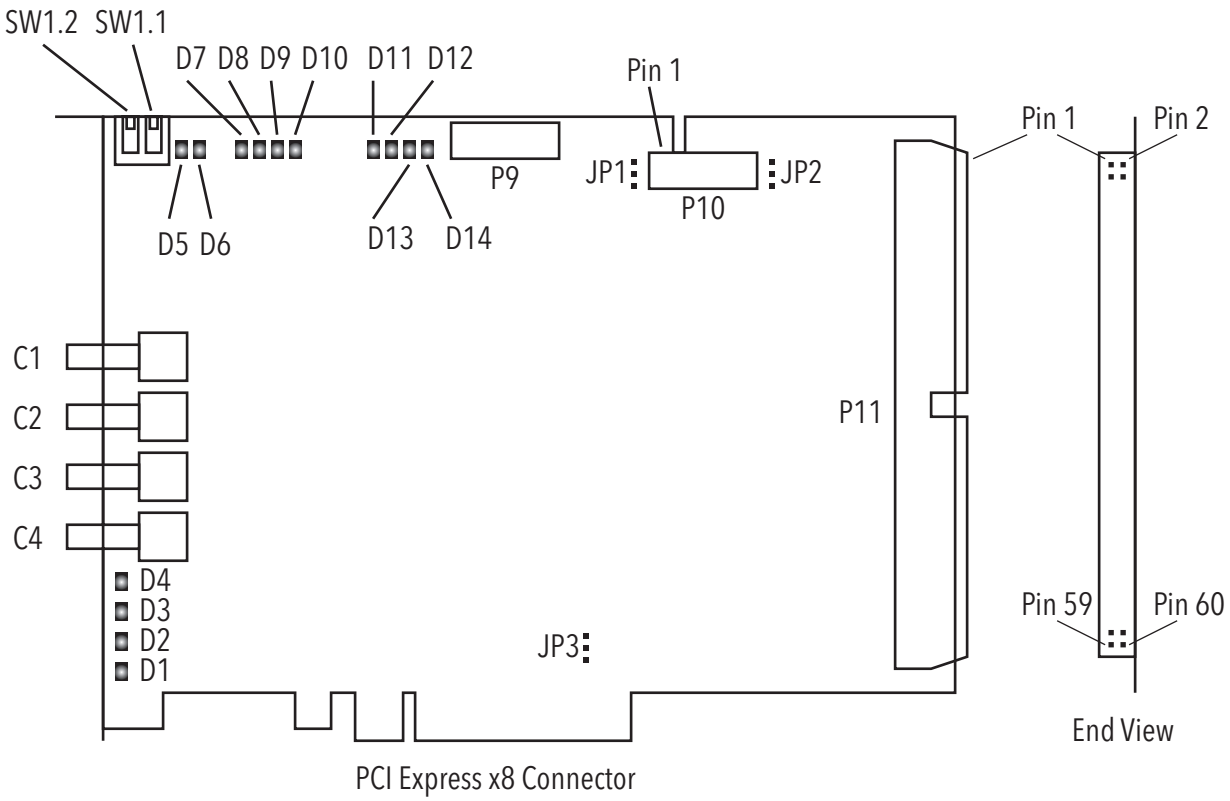

Figure 14-9 Cyton-CXP4 Board Layout (Rev 6.0 or later)

14.2 The Aon-CXP Connectors

There are four connectors on the Aon-CXP main board:

C1 - CXP connector 1
P3 - Connector for BitBox
P2 - External power connector
P6 - I/O connector

Figure 14-1 shows the locations of these connectors. The following sections show the details of each of these connectors.

14.2.1 The CXP Connector

The CXP connector is for connecting CoaXPress cameras. The Aon use DIN 1.0/2.3 connectors These are high grade 75 Ohm connectors which are compatible push/pull lock and release connectors. These connectors are fully compliant with the CoaXPress version 1.1 and later specification

14.3 The Claxon-CXP Connectors

There are eight connectors on the Claxon-CXP4/CXP2 main board:

R1 - CXP connector 1
R2 - CXP connector 2
R3 - CXP connector 3 (Claxon-CXP4 only)
R4 - CXP connector 4 (Claxon-CXP4 only)
P9 - Connector for BitBox
P10 - External power connector
P11 - I/O connector

Figure 14-2, Figure 14-3 and Figure 14-4 show the locations of these connectors. The following sections show the details of each of these connectors.

There are eight connectors on the Claxon-CXP1 main board:

R1 - CXP connector 1
P2 - Connector for BitBox
P3 - External power connector
P6 - I/O connector

Figure 14-2 shows the locations of these connectors. The following sections show the details of each of these connectors.

14.3.1 The CXP Connectors

The CXP connectors are for connecting CoaxPress cameras. The Claxon use HD BNC (also known as micro-BNC) connectors. These connectors are a more mechanically robust connection than the DIN 1.0/2.3 using on earlier models. The HD BNC also provides much better performance when used 12.5 Gb/S data rates. These connectors are fully complaint with the CoaXPress version 2.0 and later specification.

Table 14-2 illustrates how to connect the Claxon-CXP4 to various types and numbers of CoaxPress Cameras. Also for each camera, the Virtual Frame Grabber (VFG) that the camera will be connected with is shown..

Table 14-2 CXP Connector Configuration, Claxon-CXP4

Camera(s)	R1	R2	R3	R4
Single Link Cameras	Camera 1 - Link1 VFG0	Camera 2 - Link1 VFG1	Camera 3 - Link 1 VFG2	Camera 4 - Link 1 VFG3
Dual Link Cameras	Camera 1 - Link1 VFG0	Camera 1 - Link2 VFG0	Camera 2 - Link 1 VFG2	Camera 2 - Link 2 VFG2
Quad Link Camera	Camera 1 - Link1 VFG0	Camera 1 - Link2 VFG0	Camera 1 - Link 2 VFG0	Camera 1- Link 4 VFG0

14.4 The Claxon-FXP Connectors

There are four connectors on the Claxon-FXP4 main Board:

P1 - Connector for BitBox
P2 - Manufacturing Connector
P3 - I/O connector
CAGE1 - QFSP+ cage for QFSP+ fiber connectors (AOC)

Figure 14-5 shows the locations of these connectors. The following sections show the details of each of these connectors.

14.5 The Cyton-CXP Connectors

14.5.1 Cyton-CXP Rev 5.x and earlier

There are eight connectors on the Cyton-CXP4 main board:

C1 - CXP connector 1
C2 - CXP connector 2
C3 - CXP connector 3 (Cyton-CXP4 only)
C4 - CXP connector 4 (Cyton-CXP4 only)
P1 - Connector for BitBox
P4 - External power connector
P5 - I/O connector

Figure 14-6 and Figure 14-7 show the locations of these connectors. The following sections show the details of each of these connectors.

14.5.2 Cyton-CXP Rev 6.0 and later

There are eight connectors on the Cyton-CXP4 main board:

C1 - CXP connector 1
C2 - CXP connector 2
C3 - CXP connector 3 (Cyton-CXP4 only)
C4 - CXP connector 4 (Cyton-CXP4 only)
P9 - Connector for BitBox
P10 - External power connector
P11 - I/O connector

Figure 14-8 and Figure 14-9 show the locations of these connectors. The following sections show the details of each of these connectors.

14.5.3 The CXP Connectors

The CXP connectors are for connecting CoaxPress cameras. The Cyton use DIN 1.0/2.3 connectors for both the uplinks and the downlinks. These are high grade 75 Ohm connectors which are compatible push/pull lock and release connectors. These connectors offer many advantage over the BNC connectors: they use less space, they can handle higher speeds and they have better RF characteristics. These connectors are fully compliant with the CoaXPress version 1.1 and later specification.

Table 14-3 illustrates how to connect the Cyton-CXP4 to various types and numbers of CoaXPress Cameras. Also for each camera, the Virtual Frame Grabber (VFG) that the camera will be connected with is shown..

Table 14-3 CXP Connector Configuration, Cyton-CXP4

Camera(s)	C1	C2	C3	C4
Single Link Cameras	Camera 1 - Link1 VFG0	Camera 2 - Link1 VFG1	Camera 3 - Link 1 VFG2	Camera 4 - Link 1 VFG3
Dual Link Cameras	Camera 1 - Link1 VFG0	Camera 1 - Link2 VFG0	Camera 2 - Link 1 VFG2	Camera 2 - Link 2 VFG2
Quad Link Camera	Camera 1 - Link1 VFG0	Camera 1 - Link2 VFG0	Camera 1 - Link 2 VFG0	Camera 1- Link 4 VFG0

14.6 The Aon Jumpers

There is one user configurable jumper on the Aon-CXP, it controls the source of the power that is provided to the CoaXPress camera(s) connected to the Aon (PoCXP power). Table 14-4 shows the two settings.

Note: This jumper is marked "P1" on the laminate. There is another jumper on the Aon marked JP1, which serves a different function. Please see Figure 14-1 for the location of the Jumper P1.

Table 14-4 Jumper P1

Position	Meaning
Up (away from motherboard)	PoCXP power comes from the PCIe connector.
Down (closer to motherboard)	PoCXP power comes from P3.

If jumper P1 is in the Up position, then connector P3 is use to provide PoCXP power to the cameras. Contact BitFlow for adapter cables to connect internal PC power to P3.

Jumpers JP1 and JP2 are reserved and should not be changed by customers.

14.7 The Claxon Jumpers

There is two user configurable jumpers on the Cyton-CXP2/4 and one on the Claxon-CXP1. These control the source of the power that is provided to the CoaXPress camera(s) connected to the Claxon (PoCXP power). Table 14-5 shows the two settings for the Claxon-CXP2/4 and Table 14-6 shows the settings for the Claxon-CXP1.

Table 14-5 Claxon-CXP2/4 Jumper JP1 and JP2

Position	Meaning
Up (nearer the board edge)	PoCXP power comes from the PCIe connector. Use when total PoCXP power on all connectors is less than 15 watts.
Down (nearer the PCIe connector)	PoCXP power comes from P10. Use when total PoCXP power on all connectors exceeds 15 watts.

If jumper JP1 and JP2 are in the Down position, the connector P10 is use to provide PoCXP power to the cameras. See Section 14.1 for more information on this connector.

Table 14-6 Claxon-CXP1 Jumper JP1

Position	Meaning
Right (nearer the board edge)	PoCXP power comes from the PCIe connector. Use when total PoCXP power on all connectors is less than 15 watts.
Left (nearer the PCIe connector)	PoCXP power comes from P3. Use when total PoCXP power on all connectors exceeds 15 watts.

If jumper JP1 is in the Left position, the connector P3 is use to provide PoCXP power to the cameras. See Section 14.1 for more information on this connector.

Note: If the total amount of all cameras connected to the Claxon exceeds 15 Watts, then the auxiliary power connector must be used. For example, if four single link cameras each taking 4 watts are connected, then auxiliary power should be used. Similarly, if a single camera quad link camera that takes 20 wats on to links is connected, then auxiliary power must be used.

Note: In some situations, it may be desirable to connect the PoCXP power directly to the main PC power supply even if the total power draw is below 15 Watts. Tha main reason would isolation of the PoCXP power from the motherboard electronics.

14.8 The Cyton Jumpers

14.8.1 Cyton Rev 5.x and earlier

There is one user configurable jumper on the Cyton-CXP revision 5.x and earlier, it controls the source of the power that is provided to the CoaXPress camera(s) connected to the Cyton (PoCXP power). Table 14-7 shows the two settings.

Table 14-7 Jumper JP1

Position	Meaning
Left (away from P4)	PoCXP power comes from the PCIe connector. Use when total PoCXP power on all connectors is less than 15 watts.
Right (near P4)	PoCXP power comes from P4. Use when total PoCXP power on all connectors exceeds 15 watts.

If jumper JP1 is in the Right position, the connector P4 is use to provide PoCXP power to the cameras. Contact BitFlow for adapter cables to connect internal PC power to P4.

Note: If the total amount of all cameras connected to the Cyton exceeds 15 Watts, then the auxiliary power connector must be used. For example, if four single link cameras each taking 4 watts are connected, then auxiliary power should be used. Similarly, if a single camera quad link camera that takes 20 wats on to links is connected, then auxiliary power must be used.

14.8.2 Cyton Rev 6.0 and later

There are two user configurable jumper on the Cyton-CXP revision 6.0 and later, they controls the source of the power that is provided to the CoaXPress camera(s) connected to the Cyton (PoCXP power). Table 14-8 shows the two settings.

Table 14-8 Jumper JP1 and JP2

Position	Meaning
Top (away from PCIe connector)	PoCXP power comes from the PCIe connector. Use when total PoCXP power on all connectors is less than 15 watts.
Bottom (near PCIe connector)	PoCXP power comes from P104. Use when total PoCXP power on all connectors exceeds 15 watts.

If jumper JP1 and JP2 bottom position, the connector P10 is use to provide PoCXP power to the cameras. Contact BitFlow for adapter cables to connect internal PC power to P10.

Note: If the total amount of all cameras connected to the Cyton exceeds 15 Watts, then the auxiliary power connector must be used. For example, if four single link cameras each taking 4 watts are connected, then auxiliary power should be used. Similarly, if a

single camera quad link camera that takes 20 wats on to links is connected, then auxiliary power must be used.

14.9 The Aon LEDs

The Aon-CXP has 7 LEDs. Table 14-9 describes the function of these LEDS.

Note: The meanings of the LEDs where color is shown a "Various" is described in Table 14-10.

Table 14-9 Aon LEDs

LED Number	Color	Function
D1	Green	FPGA Configured
D2	Blue	General purpose, see register LED_BLUE
D3	Red	General purpose, see register LED_RED
D4	Orange	General purpose, see register LED_ORANGE
D5	Green	General purpose, see register LED_GREEN
D6	Green	Selectable VFG0 Status, see register SEL_LED
D10	Various	CXP Link 1 (C1) status

14.9.1 CXP Connector LEDs

Each CXP connector has an associated LED. The meaning of each color and blink mode is described in Table 14-10.

Table 14-10 CXP Connector LED meaning

Color	State	Meaning
Blue	Blinking	Power sense (checking for link that needs power, power is not yet applied)
Blue	Steady	Power applied (turns off after 1 second)
Green	Steady	Link established, idles being received
Green	Blinking	Packets being received
Red	Blinking	PCIe configuration problem

14.10 The Claxon LEDs

The Claxon-CXP2/4 have14 LEDs. Table 14-11 describes the function of these LEDS.

The Claxon-CXP1 has 7 LEDs. Table 14-12 describes the function of these LEDs

Note: The meanings of the LEDs where color is shown a "Various" is described in Table 14-10.

Table 14-11 Claxon-CXP2/CXP4 LEDs

LED Number	Color	Function
D1	Various	CXP Link 4 (R4) status
D2	Various	CXP Link 3 (R3) status
D3	Various	CXP Link 2 (R2) status
D4	Various	CXP Link 1 (R1) status
D5	Green	FPGA Configured
D6	Green	Reserved
D7	Green	Selectable VFG0 Status, see register SEL_LED
D8	Green	Selectable VFG1 Status, see register SEL_LED
D9	Green	Selectable VFG2 Status, see register SEL_LED
D10	Green	Selectable VFG3 Status, see register SEL_LED
D11	Blue	General purpose, see register LED_BLUE
D12	Red	General purpose, see register LED_RED
D13	Orange	General purpose, see register LED_ORANGE
D14	Green	General purpose, see register LED_GREEN

Table 14-12 Claxon-CXP1 LEDs

LED Number	Color	Function
D1	Green	FPGA Configured
D2	Blue	General purpose, see register LED_BLUE
D3	Red	General purpose, see register LED_RED
D4	Orange	General purpose, see register LED_ORANGE
D5	Green	General purpose, see register LED_GREEN
D6	Green	Selectable VFG0 Status, see register SEL_LED
D9	Various	CXP Link 1 (R1) status

14.11 The Claxon-Fiber LEDs

The Claxon-FXP/4 have14 LEDs. Table 14-13 describes the function of these LEDS.

Note: The meanings of the LEDs where color is shown a "Various" is described in Table 14-10.

Table 14-13

LED Number	Color	Function
D1	Green	FPGA Configured
D2	Green	Reserved
D3	Green	Selectable VFG0 Status, see register SEL_LED
D4	Green	Selectable VFG1 Status, see register SEL_LED
D5	Green	Selectable VFG2 Status, see register SEL_LED
D6	Green	Selectable VFG3 Status, see register SEL_LED
D7	Blue	General purpose, see register LED_BLUE
D8	Red	General purpose, see register LED_RED
D9	Orange	General purpose, see register LED_ORANGE
D10	Green	General purpose, see register LED_GREEN
D11	Various	CXP Link 1 status
D12	Various	CXP Link 2 status
D13	Various	CXP Link 3 status
D14	Various	CXP Link 4 status

14.11.1 Fiber Connector LEDs

The QFSP+ cage on the Claxon-Fiber contains four CXP fiber connections. Each connection has an associated LED. The meaning of each color and blink mode is described in Table 14-14.

Table 14-14 Fiber Connector LED meaning

Color	State	Meaning
Blue	Blinking	Power sense (checking for link that needs power, power is not yet applied)

Table 14-14 Fiber Connector LED meaning

Color	State	Meaning
Blue	Steady	Power applied (turns off after 1 second)
Green	Steady	Link established, idles being received
Green	Blinking	Packets being received
Red	Blinking	PCIe configuration problem

14.12 The Cyton LEDs

14.12.1 Cyton Rev 5.x or earlier

The Cyton-CXP has 14 LEDs. Table 14-15 describes the function of these LEDS.

Note: The meanings of the LEDs where color is shown a "Various" is described in Table 14-10.

Table 14-15 Cyton LEDs (Rev 5.x or earlier)

LED Number	Color	Function
D1	Blue	General purpose, see register LED_BLUE
D2	Red	General purpose, see register LED_RED
D3	Orange	General purpose, see register LED_ORANGE
D4	Green	General purpose, see register LED_GREEN
D5	Green	Selectable VFG0 Status, see register SEL_LED
D6	Green	Selectable VFG1 Status, see register SEL_LED
D7	Green	Selectable VFG2 Status, see register SEL_LED
D8	Green	Selectable VFG3 Status, see register SEL_LED
D9	Green	FPGA Configured
D11	Various	High speed uplink status
D14	Various	CXP Link 1 (C1) status
D15	Various	CXP Link 2 (C2) status
D16	Various	CXP Link 3 (C3) status
D18	Various	CXP Link 4 (C4) status

14.12.2 Cyton Rev 6.0 or later

The Cyton-CXP has 14 LEDs. Table 14-16 describes the function of these LEDS.

Note: The meanings of the LEDs where color is shown a "Various" is described in Table

14-10.

Table 14-16 Cyton LEDs (Rev 6.0 or later)

LED Number	Color	Function
D1	Various	CXP Link 4 (R4) status
D2	Various	CXP Link 3 (R3) status
D3	Various	CXP Link 2 (R2) status
D4	Various	CXP Link 1 (R1) status
D5	Green	FPGA Configured
D6	Green	Reserved
D7	Green	Selectable VFG0 Status, see register SEL_LED
D8	Green	Selectable VFG1 Status, see register SEL_LED
D9	Green	Selectable VFG2 Status, see register SEL_LED
D10	Green	Selectable VFG3 Status, see register SEL_LED
D11	Blue	General purpose, see register LED_BLUE
D12	Red	General purpose, see register LED_RED
D13	Orange	General purpose, see register LED_ORANGE
D14	Green	General purpose, see register LED_GREEN

14.13 The Switches

There is one piano-type switch block, SW1, with two switches. These are used to identify individual boards when there is more than one board in a system. The idea is to set the switches differently on each board in the system. The switch settings can be read for each board from software (by reading the SW bitfield). SysReg also shows the switch setting for each board. See Table 14-17 below shows the switch settings and the corresponding value in the SW bitfield.

Table 14-17 Switch S1 Setting

SW1.1	SW1.2	SW register
up	up	0
up	down	1
down	up	2
down	down	3

There is one micro switch block, SW3, on the Cyton-CXP Rev 5.x or earlier with four switches. These used to control the flash bank that the system boots from.

Note: Do not change these switches unless instructed by BitFlow support.

See Table 14-18 below which shows the switch settings and the corresponding firmware bank.

Table 14-18 Switch S3 Setting

SW3.4	SW3.2	SW 3.2	SW 3.1	FW Bank
off	off	off	off	1
off	off	off	on	2
off	off	on	off	3
off	off	on	on	4
off	on	off	off	Reserved
off	on	off	on	Reserved
off	on	on	off	Reserved
on	on	on	on	Reserved
on	off	off	off	Reserved
on	off	off	on	Reserved
on	off	on	off	Reserved
on	off	on	on	Reserved

Table 14-18 Switch S3 Setting

SW3.4	SW3.2	SW 3.2	SW 3.1	FW Bank
on	on	off	off	Reserved
on	on	off	on	Reserved
on	on	on	off	Reserved
on	on	on	on	Reserved

14.14 Button

The Cyton-CXP Rev 5.x or earlier has a general purpose button, SW2, that can be routed to many different destinations. The purpose of the button is primarily to help debug I/O problems. It can be used as a trigger, encoder, or I/O that is routed off the board. Please see Section 2.1 for more information on how the button can be routed.

14.15 The Auxiliary Power Connectors

All models normally are capable of providing power to attached CoaXPress cameras, this facility is called PoCXP. The source of this power normally comes from the PCIe bus. However, in some situations is may be desirable or necessary to use the PC's power supply to supply power. On a four-link board, if all four CXP links are taking maximum power, then the PCIe bus can not provide sufficient power and the PCs power supply must be used. On single link boards, the PCIe bus is sufficient to provide the maximum amount of PoCXP power, however, some customers may still desire to power the camera separately from the PCie bus.

It is possible to provide power to cameras that require more power than can be provide by the PCIe bus, all models have an auxiliary connector which can take power from the PC's power supply. The auxiliary power connector is accepts 12 Volts from the PCs power supply. There are a jumpers that must be in set to the correct position to route power from this connector to the CoaXPress connectors.

Note: The Auxiliary Power Connector is only used for powering the connected CXP camera, current from this connector is not use to power the components on the frame grabber.

14.15.1 Cyton (Rev 5.x and earlier)/Aon Power Connectors

The Cyton Auxiliary Power Connector is shown in Table 14-19. This connector is compatible with Berg 4-pin peripheral connectors available in many PCs. This connector is also known as the "floppy connector". For PCs that do have this type of connector, BitFlow offers an adapter cable that goes between P4 and a standard Molex 4-pin peripheral connector available in almost all PCs.

Table 14-19 Aon/Cyton Auxiliary Power Connector

Pin	Voltage
1	NC
2	GND
3	GND
4	12 Volts

14.15.2 Claxon/Cyton (Rev 6.0 and later) Power Connectors

The Cyton Auxiliary Power Connector is shown in Table 14-20. This connector is compatible with a 4-pin Molex connectors available in many PCs. This connector is also known as the peripheral power connector.

Table 14-20 Claxon/Cyton Auxiliary Power Connector

Pin	Voltage
1	12 Volts
2	GND
3	GND
4	NC

Note: The Claxon-Fiber does not provide power to the camera, so this board doe not have a auxiliary power connector.

14.16 The BitBox Connector

This connector is for connection BitFlow external I/O box, called the BitBox. Contact BitFlow for more information on the BitBox.

14.17 I/O Connector Pinout for the Cyton and Claxon

The pin-out for the I/O Connector for the Cyton-CXP and Claxon-CXP2/4 is shown in the Table 14-21.

Note: Signal names start with the Virtual Frame Grabber (VFG) that they are routed to. For example, the signal VFG0_TRIGGER_TTL is wired to VFG0., while VFG2_TRIGGER_TTL is wired to VGF2.

Table 14-21 I/O Connector for the Cyton and Claxon

Pin	I/O	Signal	Comment
1	In	VFG0_TRIGGER+	LVDS
2	In	VFG0_TRIGGER-	LVDS
3	In	VFG0_ENCODERA+	LVDS
4	In	VFG0_ENCODERA-	LVDS
5	In	VFG0_ENCODERB+	LVDS
6	In	VFG0_ENCODERB-	LVDS
7	In	VFG1_TRIGGER+	LVDS
8	In	VFG1_TRIGGER-	LVDS
9	In	VFG1_ENCODERA+	LVDS
10	In	VFG1_ENCODERA-	LVDS
11	In	VFG1_ENCODERB+	LVDS
12	In	VFG1_ENCODERB-	LVDS
13	In	VFG2_TRIGGER+	LVDS
14	In	VFG2_TRIGGER-	LVDS
15	In	VFG2_ENCODERA+	LVDS
16	In	VFG2_ENCODERA-	LVDS
17	In	VFG2_ENCODERB+	LVDS
18	In	VFG2_ENCODERB-	LVDS
19	In	VFG3_TRIGGER+	LVDS
20	In	VFG3_TRIGGER-	LVDS
21	In	VFG3_ENCODERA+	LVDS
22	In	VFG3_ENCODERA-	LVDS
23	In	VFG3_ENCODERB+	LVDS
24	In	VFG3_ENCODERB-	LVDS
25		GND	
26	Out	VFG0_CC3+	LVDS
27	Out	VFG0_CC3-	LVDS
28	Out	VFG1_CC3+	LVDS
29	Out	VFG1_CC3-	LVDS
30	Out	VFG2_CC3+	LVDS
31	Out	VFG2_CC3-	LVDS

Table 14-21 I/O Connector for the Cyton and Claxon

Pin	I/O	Signal	Comment
32	Out	VFG3_CC3+	LVDS
33	Out	VFG3_CC3-	LVDS
34		GND	
35	In	VFG0_TRIGGER_TTL	TTL
36	In	VFG0_ENCODERA_TTL	TTL
37	In	VFG0_ENCODERB_TTL	TTL
38	In	VFG1_TRIGGER_TTL	TTL
39	In	VFG1_ENCODERA_TTL	TTL
40	In	VFG1_ENCODERB_TTL	TTL
41	In	VFG2_TRIGGER_TTL	TTL
42	In	VFG2_ENCODERA_TTL	TTL
43	In	VFG2_ENCODERB_TTL	TTL
44	In	VFG3_TRIGGER_TTL	TTL
45	In	VFG3_ENCODERA_TTL	TTL
46	In	VFG3_ENCODERB_TTL	TTL
47		Reserved	
48	Out	VFG0_CC3_TTL	TTL
49	Out	VFG0_CC4_TTL	TTL
50	Out	VFG0_CC2_TTL	TTL
51	Out	VFG1_CC3_TTL	TTL
52	Out	VFG1_CC4_TTL	TTL
53	Out	VFG1_CC2_TTL	TTL
54	Out	VFG2_CC3_TTL	TTL
55	Out	VFG2_CC4_TTL	TTL
56	Out	VFG2_CC2_TTL	TTL
57	Out	VFG3_CC3_TTL	TTL
58	Out	VFG3_CC4_TTL	TTL
59	Out	VFG3_CC2_TTL	TTL
60		GND	

14.18 I/O Connector Pinout for the Aon-CXP and Claxon-CXP1

The pin-out for the I/O Connector for the Aon-CXP and Claxon-CXP1 is shown in the Table 14-22.

Table 14-22 I/O Connector for the Aon-CXP and Claxon-CXP1

Pin	I/O	Signal	Comment
1	In	VFG0_TRIGGER-	LVDS
2	In	VFG0_ENCODERA-	LVDS
3	In	VFG0_ENCODERB-	LVDS
4	Out	VFG0_CC3-	LVDS
5	In	VFG0_ENCODERA_TTL	TTL
6	In	VFG0_TRIGGER+	LVDS
7	In	VFG0_ENCODERA+	LVDS
8	In	VFG0_ENCODERB+	LVDS
9	Out	VFG0_CC3+	LVDS
10	In	VFG0_TRIGGER_TTL	TTL
11	In	VFG0_ENCODERB_TTL	TTL
12	Out	VFG0_CC2_TTL	TTL
13	Out	VFG0_CC3_TTL	TTL
14	Out	VFG0_CC4_TTL	TTL
15		GND	

Index

Numerics

A

B

C

D

E

F

I

J

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z