

# **The Cyton-CXP and Aon-CXP**

## **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](mailto:support@bitflow.com)  
Web: [www.bitflow.com](http://www.bitflow.com)  
Revision B.0

© 2017 BitFlow, Inc. All Rights Reserved.

This document, in whole or in part, may not be copied, photocopied, reproduced, translated or reduced to any other electronic medium or machine readable form without the prior written consent of BitFlow, Inc.

BitFlow, Inc. makes no implicit warranty for the use of its products and assumes no responsibility for any errors that may appear in this document, nor does it make a commitment to update the information contained herein.

BitFlow, Inc. retains the right to make changes to these specifications at any time without notice.

All trademarks are properties of their respective holders.

Revision History:

<b>Revision</b>	<b>Date</b>	<b>Comments</b>
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

# Table of Contents

---

## P - Preface

- Purpose CYT-P-1
  - Support Services CYT-P-1
  - Technical Support CYT-P-1
  - Sales Support CYT-P-2
  - Conventions CYT-P-2
- Bitfield definitions CYT-P-3
  - Example Bitfield Definition CYT-P-3
  - Bitfield Definition Explanation. CYT-P-3

## 1 - General Description and Architecture

- The Cyton-CXP and Aon-CXP families CYT-1-1
  - CoaXPress CYT-1-1
  - Virtual vs. Hardware CYT-1-1
  - The Virtual Frame Grabber (VFG) CYT-1-2
  - Configuration Spaces CYT-1-2
- General Description CYT-1-5
  - Video Data CYT-1-6
  - Command and Control Packets CYT-1-7
  - CXP Triggers CYT-1-7
  - Uplink CYT-1-7
  - Cyton/Aon-CXP I/O system CYT-1-7
  - The Timing Sequencer Signal Generator CYT-1-7
  - The StreamSync System CYT-1-8
  - The Volume Of Interest Acquisition Engine CYT-1-8
  - CoaXPress Power CYT-1-8
- Firmware CYT-1-10
- Cyton/Aon Camera Configuration Files CYT-1-11
  - BFML Camera File Modes CYT-1-11
- The Routing of CXP Links to VFGs on the Cyton-CXP CYT-1-12
- The BitFlow CXP Models CYT-1-13

## 2 - The StreamSync Acquisition Engine

- Introduction CYT-2-1
- The StreamSync Acquisition Engine World CYT-2-2
  - Controlling the StreamSync Acquisition Engine CYT-2-2
  - Observing the StreamSync Acquisition Engine CYT-2-4
  - Synchronizing the StreamSync Acquisition Engine With a CXP Camera CYT-2-4
  - Regions Of Interest (ROI) with the StreamSync Acquisition Engine. CYT-2-4
- Triggering the StreamSync Acquisition Engine CYT-2-6
- Comparing the StreamSync Acquisition Engine to Other BitFlow products CYT-2-7

AE\_CON CYT-2-8  
AE\_STATUS CYT-2-10  
AE\_STREAM\_SEL CYT-2-12  
V\_WIN\_DIM CYT-2-14  
Z\_WIN\_CON CYT-2-16  
Z\_WIN\_DIM CYT-2-20  
Z\_WIN\_DIM\_EXT CYT-2-22  
Y\_WIN\_CON CYT-2-24  
Y\_WIN\_DIM CYT-2-28  
Y\_WIN\_DIM\_EXT CYT-2-30  
X\_WIN\_DIM CYT-2-32  
V\_ACQUIRED CYT-2-34  
Z\_ACQUIRED CYT-2-36  
Y\_ACQUIRED CYT-2-38  
X\_ACQUIRED CYT-2-40  
CON489 CYT-2-42  
CON490 CYT-2-45  
CON548 CYT-2-47  
CON549 CYT-2-50  
SF\_DIM CYT-2-53  
SF\_CON CYT-2-55

### **3 - The StreamSync Buffer Manager**

Introduction CYT-3-1  
The Buffer Manager Details CYT-3-2  
CON485 Register CYT-3-3  
CON486 Register CYT-3-5  
BUF\_MGR\_CON CYT-3-7  
BUF\_MGR\_TIMEOUT CYT-3-9  
BOARD\_CONFIG CYT-3-11  
PACKETS\_SENT\_STATUS CYT-3-13  
QUADS\_USED\_STATUS CYT-3-15  
QTABS\_USED\_STATUS CYT-3-17  
PKT\_STAT CYT-3-19  
QUADS\_LOADED\_STATUS CYT-3-22  
QTABS\_LOADED\_STATUS CYT-3-24  
BUF\_MGR\_STATUS CYT-3-26  
PKT\_CON CYT-3-29

### **4 - Timing Sequencer**

Introduction CYT-4-1  
    The Auxiliary Timing Sequencer CYT-4-1  
    Description CYT-4-1  
TS\_CONTROL CYT-4-3  
TS\_TABLE\_CONTROL CYT-4-6  
TS\_TABLE\_ENTRY CYT-4-8

ATS\_CONTROL CYT-4-11  
ATS\_TABLE\_CONTROL CYT-4-14  
ATS\_TABLE\_ENTRY CYT-4-16

## **5 - The Aon, Axion And Cyton I/O System**

Introduction CYT-5-1  
    Concepts CYT-5-1  
    I/O Between Virtual Frame Grabbers CYT-5-1  
Overview of the Aon, Axion and Cyton I/O System Routing CYT-5-2  
Input Selection CYT-5-3  
Internal Signals CYT-5-4  
Output Signal Selection CYT-5-7  
I/O Connector Output Signal Routing CYT-5-8  
BitBox Output Signal Routing CYT-5-9

## **6 - The Aon, Axion and Cyton I/O System Registers**

Introduction CYT-6-1  
CON60 CYT-6-2  
CON61 CYT-6-4  
CON62 CYT-6-6  
CON63 CYT-6-10  
CON64 CYT-6-15  
ADDR\_TRIG\_FILTER CYT-6-19  
ADDR\_ENCA\_FILTER CYT-6-21  
ADDR\_ENCB\_FILTER CYT-6-23  
BOX\_OUT\_DYN\_SEL\_SET\_A CYT-6-25  
BOX\_OUT\_DYN\_SEL\_SET\_B CYT-6-28  
BOX\_OUT\_DYN\_SEL\_SET\_C CYT-6-30  
BOX\_OUT\_MODE\_SET\_A CYT-6-32  
BOX\_OUT\_MODE\_SET\_B CYT-6-37  
BOX\_OUT\_MODE\_SET\_C CYT-6-42

## **7 - Encoder Divider**

Introduction CYT-7-1  
Encoder Divider Details CYT-7-2  
    Formula CYT-7-2  
    Example CYT-7-2  
    Restrictions CYT-7-2  
    PLL Locking CYT-7-3  
    Handling Encoder Slow Down or Stopping CYT-7-3  
Encoder Divider Control Registers CYT-7-4

## **8 - Quadrature Encoder**

Introduction	CYT-8-1
Simple Encoder Mode	CYT-8-1
Positive or Negative Only Acquisition	CYT-8-1
Interval Mode	CYT-8-2
Re-Acquisition Prevention	CYT-8-2
Scan Step Mode	CYT-8-2
Combining Modes	CYT-8-2
Control Registers	CYT-8-2
Observability	CYT-8-3
Electrical Connections	CYT-8-3
Understanding Stage Movement vs. Quadrature Encoder Modes	CYT-8-4

## **9 - Quadrature Encoder and Divider Registers**

Introduction	CYT-9-1
CON65 Register	CYT-9-2
CON66 Register	CYT-9-4
CON67 Register	CYT-9-8
CON68 Register	CYT-9-11
CON69 Register	CYT-9-13

## **10 - CXP Subsystem Registers**

Introduction	CYT-10-1
CON104	CYT-10-2
CON105	CYT-10-6
CON106	CYT-10-8
CON107	CYT-10-10
CON108	CYT-10-12
CON109	CYT-10-14
CON110	CYT-10-16
CON111	CYT-10-18
CON115	CYT-10-20
CON116	CYT-10-22
CON117	CYT-10-26
CON118	CYT-10-29
CON120	CYT-10-32
CON121	CYT-10-34
CON122	CYT-10-36
CON123	CYT-10-38
CON127	CYT-10-40
CON128	CYT-10-42
CON131	CYT-10-44
CON136	CYT-10-46
CON137	CYT-10-49
CON138	CYT-10-51

## Table of Contents

CON139	CYT-10-53
CON140	CYT-10-55
CON141	CYT-10-57
CON142	CYT-10-59
CON143	CYT-10-61
CON147	CYT-10-63
CON148	CYT-10-65
CON149	CYT-10-68
CON150	CYT-10-71
CON152	CYT-10-74
CON153	CYT-10-76
CON154	CYT-10-78
CON155	CYT-10-80
CON159	CYT-10-82
CON160	CYT-10-84
CON163	CYT-10-86
CON168	CYT-10-88
CON169	CYT-10-91
CON170	CYT-10-93
CON171	CYT-10-95
CON172	CYT-10-97
CON173	CYT-10-99
CON174	CYT-10-101
CON175	CYT-10-103
CON179	CYT-10-105
CON180	CYT-10-107
CON181	CYT-10-110
CON182	CYT-10-113
CON184	CYT-10-116
CON185	CYT-10-118
CON186	CYT-10-120
CON187	CYT-10-122
CON191	CYT-10-124
CON192	CYT-10-126
CON195	CYT-10-128
CON200	CYT-10-130
CON201	CYT-10-133
CON202	CYT-10-135
CON203	CYT-10-137
CON204	CYT-10-139
CON205	CYT-10-141
CON206	CYT-10-143
CON207	CYT-10-145
CON211	CYT-10-147
CON212	CYT-10-149
CON213	CYT-10-152
CON214	CYT-10-155
CON216	CYT-10-158
CON217	CYT-10-160
CON218	CYT-10-162

CON218 CYT-10-164  
CON219 CYT-10-166  
CON223 CYT-10-168  
CON224 CYT-10-170  
CON227 CYT-10-172  
CON356 CYT-10-174  
CON357 CYT-10-176  
CON358 CYT-10-178  
CON360 CYT-10-180  
CON361 CYT-10-182  
CON363 CYT-10-184  
CON372 CYT-10-186  
CON428 CYT-10-188

## 11 - System Probe

Introduction Manual-11-1  
Example - Measure CL Clock Frequency Manual-11-2  
Example - Measure LVALs Per Frame Manual-11-3  
List of Events Manual-11-4  
SP\_EVENTS Manual-11-6  
SP\_CON Manual-11-9  
SP\_STAT Manual-11-12  
SP\_LIMIT Manual-11-14

## 12 - Specifications

Introduction CYT-12-1  
PCI Express Compatibility CYT-12-3  
Power Requirements CYT-12-4

## 13 - Mechanical

Introduction CYT-13-1  
The Cyton-CXP Connectors CYT-13-4  
    The CXP Connectors CYT-13-4  
The Aon-CXP Connectors CYT-13-5  
    The CXP Connector CYT-13-5  
The Cyton Switches CYT-13-6  
The Aon Switches CYT-13-8  
The Cyton Jumpers CYT-13-9  
The Aon Jumpers CYT-13-10  
The Cyton LEDs CYT-13-11  
    CXP Downlink LED Meaning CYT-13-12  
    CXP Uplink LED Meaning CYT-13-12  
The Aon LEDs CYT-13-13  
Button CYT-13-14



## Table of Contents

The Auxiliary Power Connector (Cyton P4, Aon P2 )	CYT-13-15
The BitBox Connector (Cyton P1, Aon P3)	CYT-13-16
I/O Connector Pinout for the Cyton-CXP P5	CYT-13-17
I/O Connector Pinout for the Aon-CXP P6	CYT-13-19



# Preface

---

## Chapter P

### P.1 Purpose

This Hardware Reference Manual is intended for anyone using the Cyton-CXP or Aon-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-CXP is almost identical to the Cyton-CXP. The major difference is that the Aon-CXP has one CXP link and one VFG, while the Cyton-CXP has four CXP links and four VFGs. Thus wherever a feature is listed for the Cyton-CXP, this feature will also be available on the Aon-CXP. In cases where these two products are different, 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](http://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](mailto:support@bitflow.com)) or by FAX (781-933-9965), please include the following:

- Product name
- Camera type and mode being used
- Software revision number
- 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](http://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**

<b>Base</b>	<b>Designator</b>	<b>Example</b>
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**

<b>Abbreviation</b>	<b>Value</b>	<b>Example</b>
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, 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 written. 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**

<b>Access</b>	<b>Meaning</b>
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
R64	This bitfield is functional on the R64 family.
Alta	This bitfield is functional on the Alta family.
Cyton-CXP	This bitfield is functional on the Cyton-CXP family
Axion-CL	This bitfield is functional on the Axion-CL family
Aon-CXP	This bitfield is functional on the Aon-CL family

# General Description and Architecture

---

## Chapter 1

### 1.1 The Cyton-CXP and Aon-CXP families

The purpose of this chapter is to explain, at a block diagram level, how the Cyton-CXP and Aon-CXP work. Currently there is two main models in the Cytron-CXP family:

CYT-PC2-CXP4, provides four 6.25 Gb/S CXP links

CYT-PC2-CXP2, provides two 6.25 Gb/S CXP links

And there is one model of the Aon-CXP:

AON-PC2-CXP1, provides one 6.25 Gb/S CXP link

#### 1.1.1 CoaXPress

In order to understand how the Cyton-CXP and Aon-CXP 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](http://www.coaxpress.com).

#### 1.1.2 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 hardware 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 Cyton-CXP, 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.

### 1.1.3 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.

### 1.1.4 Configuration Spaces

The Cyton-CXP model supports up to four 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. Figure 1-1 shows the block diagram of the for Cyton.

The Aon-CXP has only one configuration space, and thus has only one VFG. Figure 1-2 shows the block diagram for the Aon.



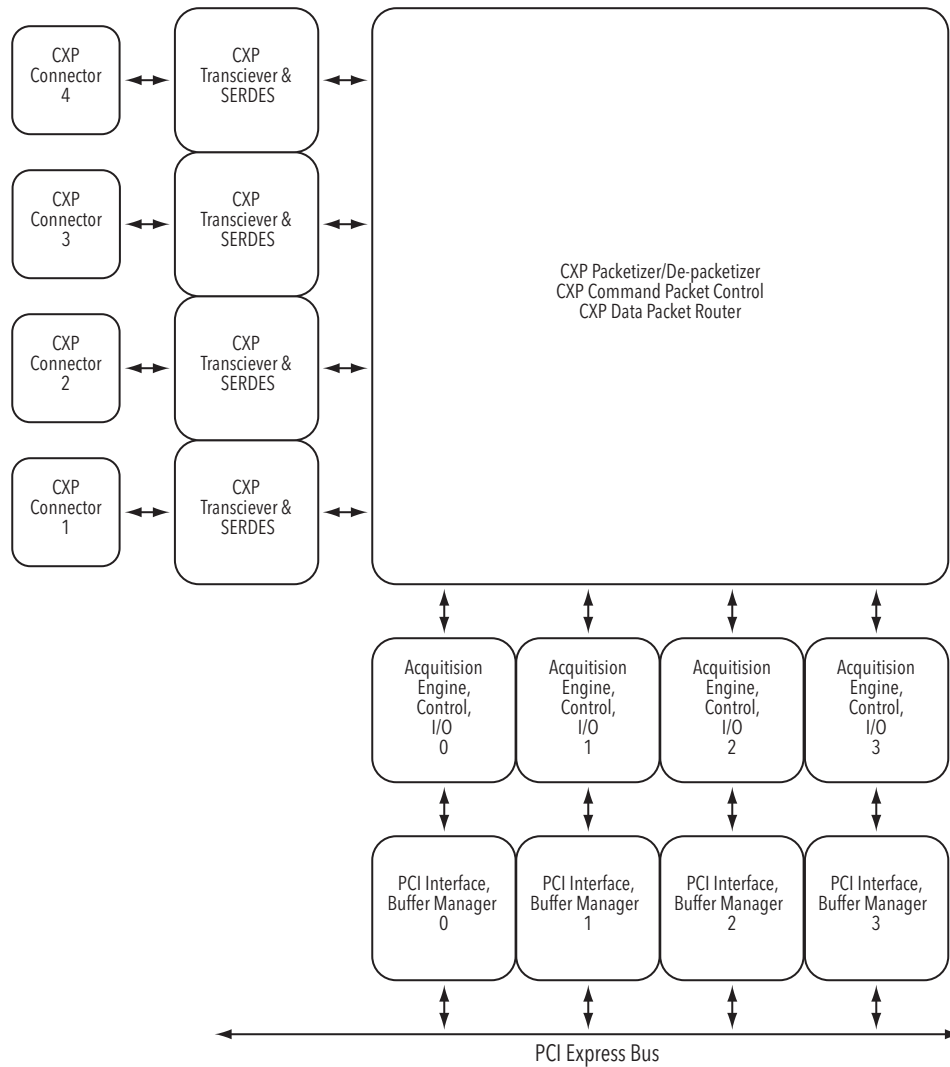


Figure 1-1 The Cyton-CXP Block Diagram

Note: The Cyton-CXP2 only has two CXP channels and two PCI interfaces.

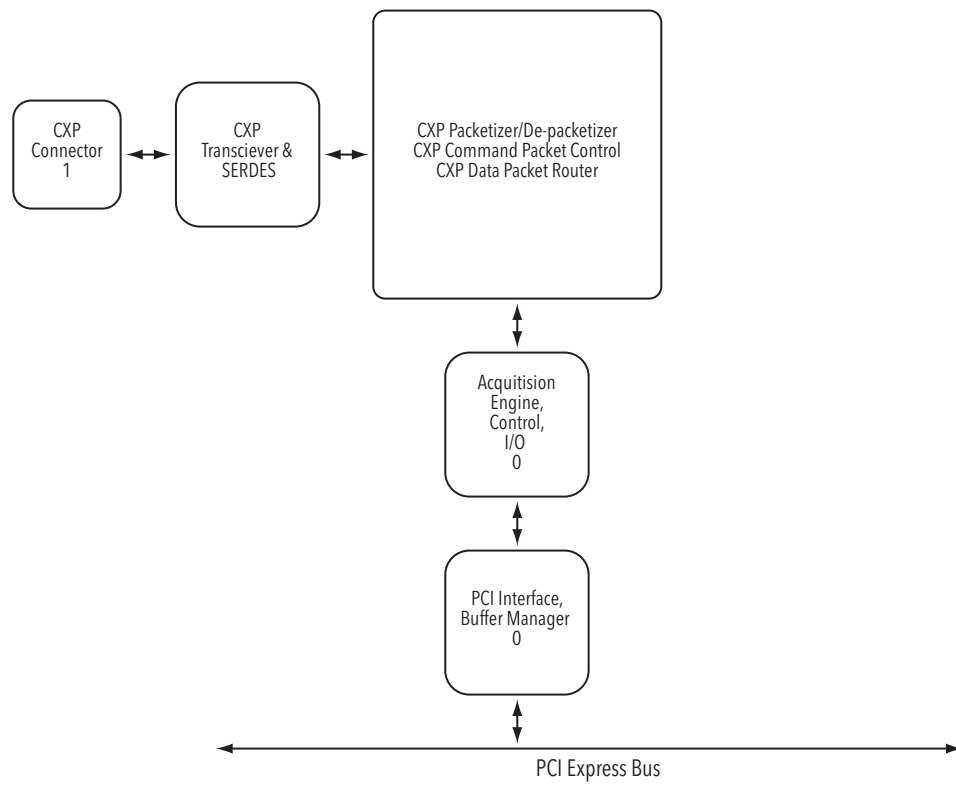


Figure 1-2 The Aon-CXP Block Diagram

## 1.2 General Description

The Cyton-CXP is a x8 PCI Express Gen 2 board. It can work in any PCI Express slot that it can fit in. Usually this means an x8 or x16 slot. However, some mother boards have x4 slots with x8 connectors. The Cyton-CXP will work in these slots, although performance may be somewhat reduced. The Cyton-CXP is a Gen 2 PCIe device, but it will work in Gen 1 slots, though DMA performance will be degraded. DMA performance will be the same for both Gen 2 and Gen 3 slots.

*Note: As of SDK 6.2, the Cyton will only work in PCIe x8 and x16 slots that are Gen 2 or Gen 3. BitFlow is working on support for other slot sizes and generations.*

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

The Cyton-CXP is extremely flexible. Each VFG can acquire from any of the CXP links on the board (although certain restrictions usually apply). In addition, each VFG can acquire from one or more CXP links. This flexibility means that, for example, the Cyton-CXP4 can be used as two VFGs, each acquiring from a dual link CXP camera, or one VFG acquiring from a quad link CXP camera (in this case, the other VFGs would not be used).

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

Figure 1-3 shows the block diagram of a Cyton-CXP VFG when it is used with a single link camera. Note that this is also the block diagram of the VFG on the Aon-CXP.

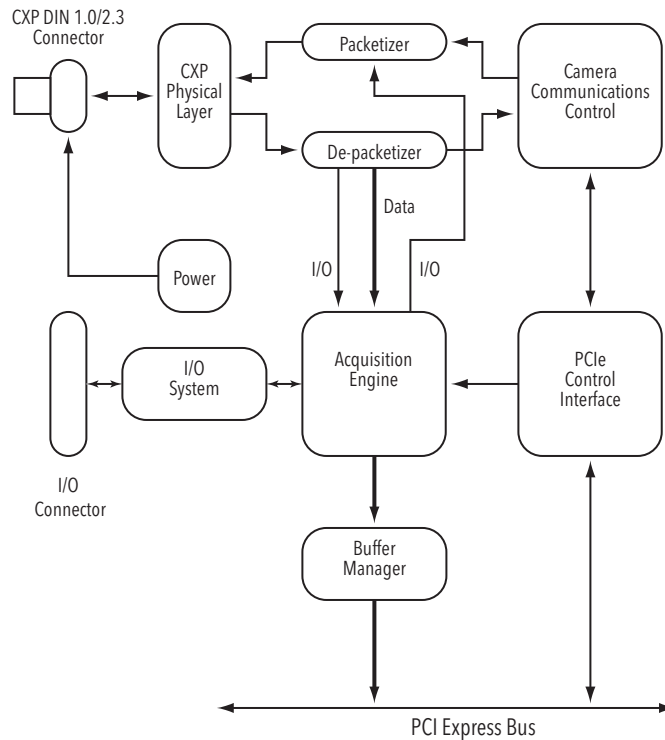


Figure 1-3 Cyton-CXP Single Link Configuration and the Aon-CXP Block Diagram

### 1.2.1 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 used to read/write from/to the camera's 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 1-3). From this point on the Cyton/Aon-CXP 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 is read by the DMA engine. The DMA engine then DMA's 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.

## 1.2.2 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 of the CoaXPress specification.

## 1.2.3 CXP Triggers

The third type of packet the Cyton-CXP handles 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.

## 1.2.4 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.

## 1.2.5 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.

## 1.2.6 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-CXP provides the same TS as the Cyton-CXP.

### 1.2.7 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-CXP use the same StreamSync Acquisition engine as the Cyton.

### 1.2.8 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.

### 1.2.9 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 Cyton/Aon-CXP conforms to this specification. Some cameras do not require power, so the Cyton/Aon-CXP 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 Cyton/Aon-CXP 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 Cyton/Aon-CXP 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 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.*

*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.*

## 1.3 Firmware

Unlike many of BitFlow's previous models of frame grabbers, the Cyton and families do not swap firmware on the fly. The Cyton and Aon 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 Cyton/Aon 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-CXP uses the same format camera configuration file as the Cyton-CXP. A BFML configuration file for a single link camera will work with both the Aon-CXP and the Cyton-CXP.

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 Cyton-CXP

What Figure 1-3 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 1-4 shows the all the theoretical routing possibilities.

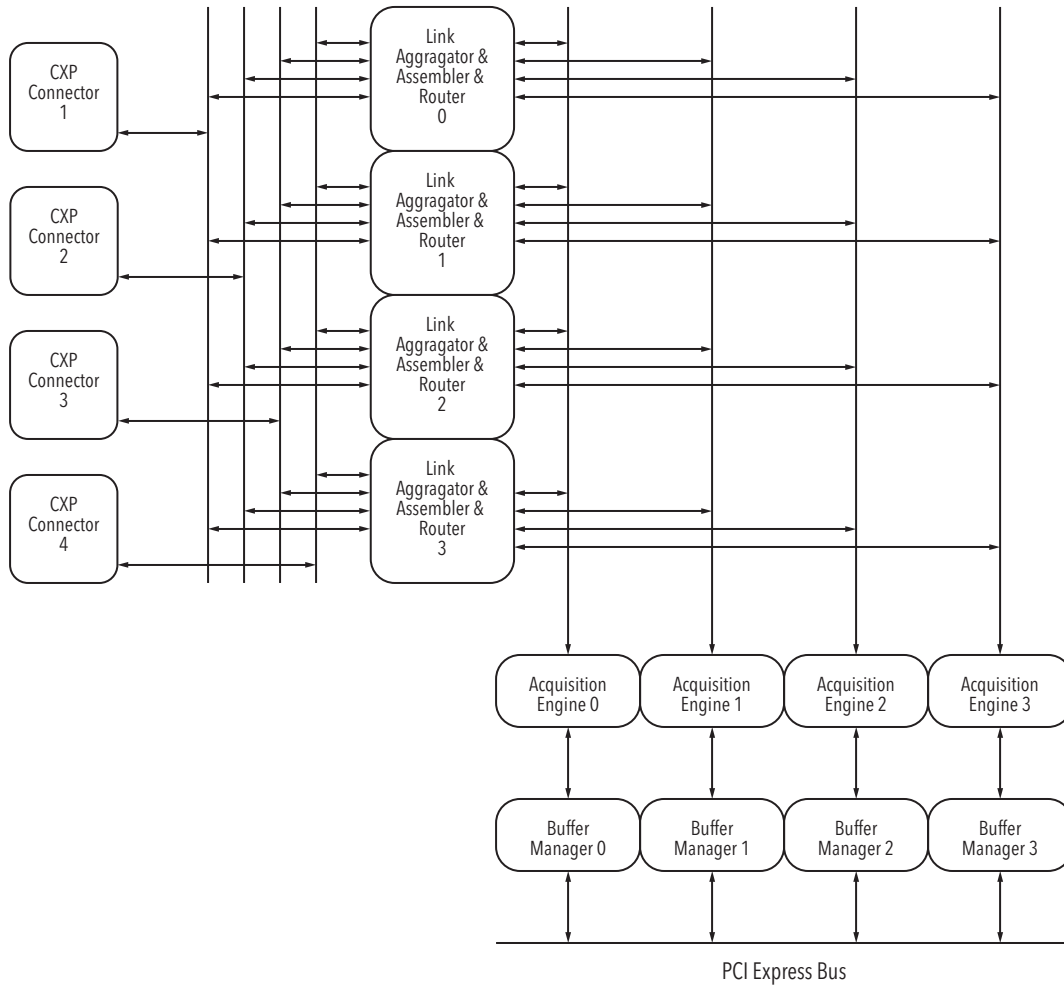


Figure 1-4 Cyton-CXP Routing Between CXP Links and VFG

Figure 1-4 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	CYT-PC2-CXP4	CYT-PC2-CXP4	AON-PC2-CXP1
Number of 3.125 Gb/S links supported	4	2	1
Number of 6.25 Gb/S links supported	4	2	1
Number of single link cameras supported	4	2	1
Number of dual link cameras supported	2	1	0
Number of quad link cameras supported	1	0	0
Number of Virtual Frame Grabbers supported	4	2	1
Number of independent trigger inputs	4	2	1
Number of independent encoder inputs	4	2	1
Number of PCI configurations (devices)	4	2	1
Maximum DMA bandwidth	3.5 GB/S	3.5 GB/S	1.5 GB/S



# 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. 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 and the Cyton-CXP.

From a software perspective, the StreamSync system is compatible with the previous BitFlow products. However, digging deeper, these new systems 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 concept that an image has 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.

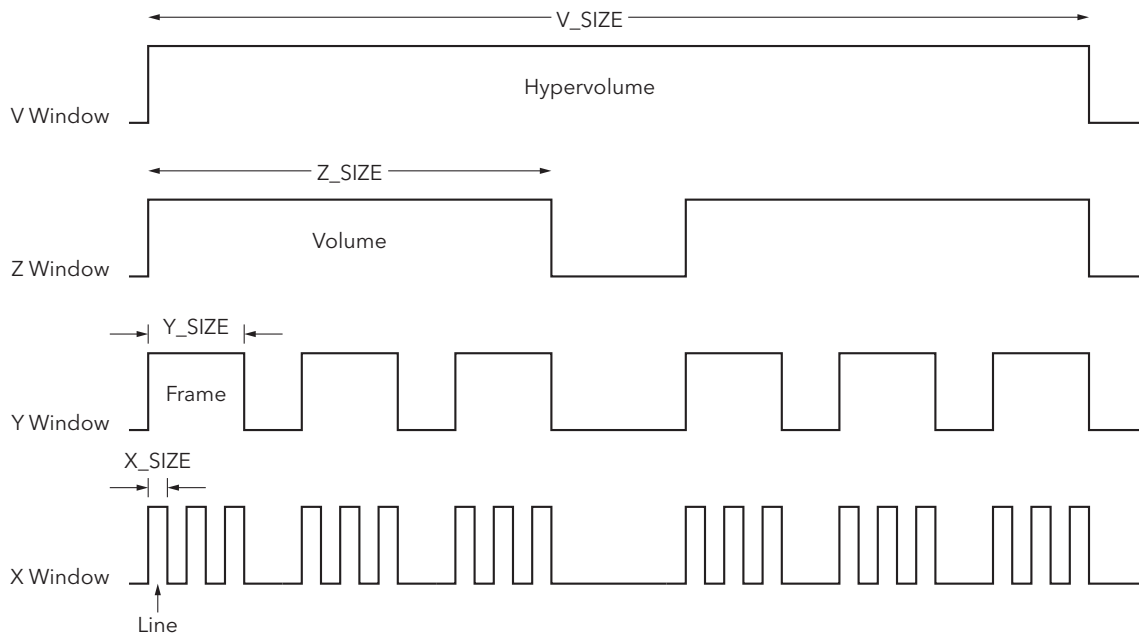


Figure 2-1 StreamSync Acquisition Engine Dimensions

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:

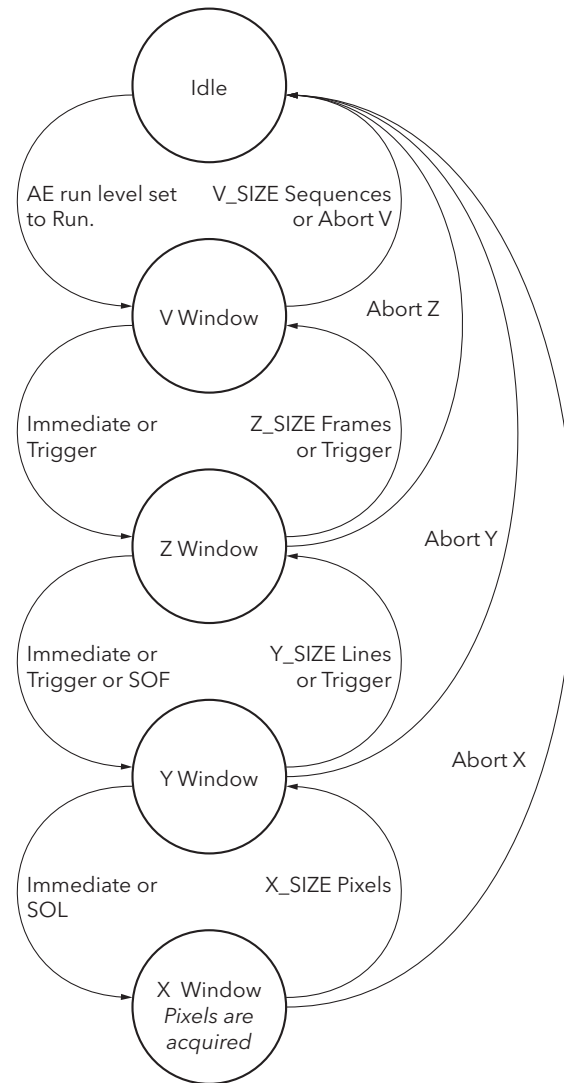


Figure 2-2 Acquisition Engine Run Level

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 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 Z Windows
Z	Trigger, Immediate	Trigger, Abort Z, Z_SIZE Y Windows
Y	Trigger, SOF, Immediate	Trigger, Abort Y, Y_SIZE X Windows
X	SOL, 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).

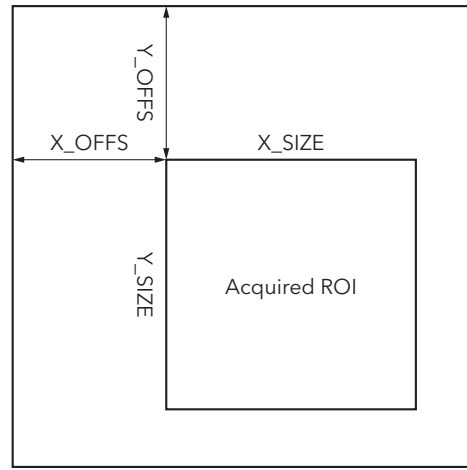
## 2.2.3 Synchronizing the StreamSync Acquisition Engine With a CXP Camera

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 synchronize to the camera even if packets are dropped. This functionality can be enable/disable by the ZSYNC and YSYNC bitfields.

## 2.2.4 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 t are Y\_OFFS and X\_OFFS registers which can locate the subwindow anyw inside of the camera's frame. These concepts are show in Figure 2-3.

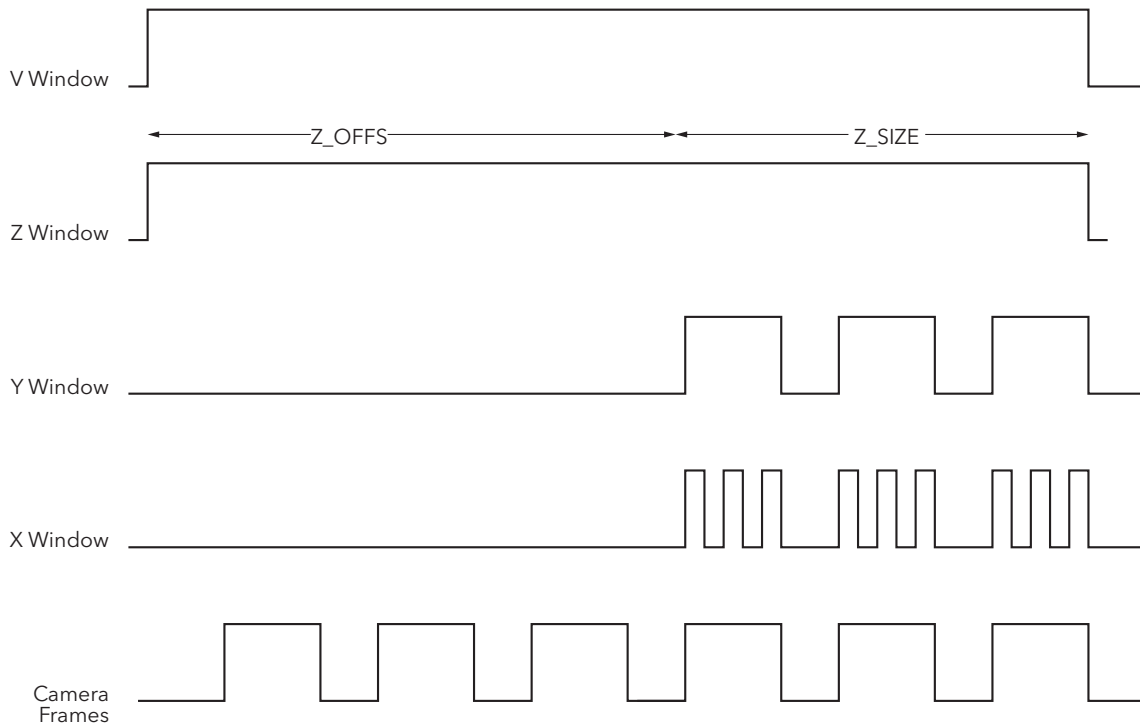




Camera's Frame

**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 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 used 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 start with one trigger or end 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.

## 2.4 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 code, 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 traditional 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	0xffff	Run
Grab N frames	Camera Width	Camera Height	N	1	Run
Freeze					Abort Z
Abort					Abort X

## 2.5 AE\_CON

<b>Bit</b>	<b>Name</b>
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	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\_RUN\_LEVEL** R/W, AE\_CON[3..0], Aon-CXP, Axion-CL, Cyton-CXPL

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.

<b>AE_RUN_LEVEL</b>	<b>Meaning</b>
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

## 2.6 AE\_STATUS

<b>Bit</b>	<b>Name</b>
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, Cyton-CXP

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

<b>AE_STATE</b>	<b>Meaning</b>
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, Cyton-CXP

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

## 2.7 AE\_STREAM\_SEL

<b>Bit</b>	<b>Name</b>
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, Cyton-CXP

Program this register to the stream aggregator that this acquisition engine should get its data from. Currently only value 0 to 3 are supported. Generally this register should be programmed to correspond to the VFG number that is being use 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, Cyton-CXP  
Use the Synthetic Frame generator instead of the camera.

## 2.8 V\_WIN\_DIM

<b>Bit</b>	<b>Name</b>
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, 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.

## 2.9 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, 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, 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, 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 there 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 there are more volumes to acquire.

**Z\_OPEN\_TRIG\_FUNC**

R/W, Z\_WIN\_CON[15..12], Aon-CXP, Axion-CL, 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, 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, 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, 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.

<b>Z_SYNC</b>	<b>Meaning</b>
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 streaming packets received before the SOF will be ignored. This conditions is enforced for each frame in the Z window.

## 2.10 Z\_WIN\_DIM

<b>Bit</b>	<b>Name</b>
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, 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, 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.

## 2.11 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, 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, Cyton-CXP  
Extends the Z\_OFFS register by 8 more bits.

## 2.12 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, 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, Cyton-CXP

Selects which trigger will control the end the 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\_CLOSE** R/W, Y\_WIN\_CON[11..8], Aon-CXP, Axion-CL, 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\_SIZEZ\_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, 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, 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, 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, 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 shows explains this field.

<b>Y_SYNC</b>	<b>Meaning</b>
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 streaming packets received before the SOL will be ignored. This conditions is enforced for each line in the Y window.

## 2.13 Y\_WIN\_DIM

<b>Bit</b>	<b>Name</b>
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, 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, Cyton-CXP

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

## 2.14 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, 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, Cyton-CXP  
Extends the Y\_OFFS register by 8 more bits.

## 2.15 X\_WIN\_DIM

<b>Bit</b>	<b>Name</b>
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, 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, Cyton-CXP

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

## 2.16 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	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	V_ACQ_COUNT_CLR_MODE
29	V_ACQ_COUNT_CLR_MODE
30	V_ACQ_COUNT_CLR_MODE
31	Reserved

**V\_ACQ\_COUNT** R/W, V\_ACQUIRED[15..0], Aon-CXP, Axion-CL, 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 its 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, Cyton-CXP

Controls how the V\_ACQ\_COUNT register is cleared.

V_ACQ_COUNT_CLEAR_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

## 2.17 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	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Z_ACQ_COUNT_CLR_MODE
29	Z_ACQ_COUNT_CLR_MODE
30	Reserved
31	Reserved



**Z\_ACQ\_COUNT** R/W, Z\_ACQUIRED[15..0], Aon-CXP, Axion-CL, Cyton-CXP

Returns the total number of frames acquired since the last reset of this register. The behavior of this register when it reaches its 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, Cyton-CXP

Controls how the Z\_ACQ\_COUNT register is cleared.

Z_ACQ_COUNT_CLEAR_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

## 2.18 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	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_ACQ_COUNT_CLR_MODE
29	Y_ACQ_COUNT_CLR_MODE
30	Reserved
31	Reserved

**Y\_ACQ\_COUNT** R/W, Y\_ACQUIRED[15..0], Aon-CXP, Axion-CL, Cyton-CXP

Returns the total number of lines acquired since the last reset of this register. The behavior of this register when it reaches its 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, Cyton-CXP

Controls how the Y\_ACQ\_COUNT register is cleared.

Y_ACQ_COUNT_CLEAR_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

## 2.19 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	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	X_ACQ_COUNT_CLR_MODE
29	X_ACQ_COUNT_CLR_MODE
30	Reserved
31	Reserved

**X\_ACQ\_COUNT** R/W, X\_ACQUIRED[15..0], Aon-CXP, Axion-CL, Cyton-CXP

Returns the total number of 16-bit words acquired since the last reset of this register. The behavior of this register when it reaches its 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, Cyton-CXP

Controls how the X\_ACQ\_COUNT register is cleared.

<b>X_ACQ_COUNT_CLEAR_MODE</b>	<b>Meaning</b>
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

## 2.20 CON489

Bit	Name
0	INT_Z_ACQUIRED
1	INT_Y_ACQUIRED
2	INT_V_ACQUIRED
3	Reserved
4	INT_ENC_B
5	INT_ENC_A
6	INT_TRIG
7	INT_Z_START
8	INT_Y_START
9	INT_V_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_Z_ACQUIRED_LEGACY
30	Reserved
31	Reserved

<b>INT_Z_ACQUIRED</b>	R/W, CON489[0], Aon-CXP, Axion-CL, Cyton-CXP Z window closed interrupt.
<b>INT_Y_ACQUIRED</b>	R/W, CON489[1], Aon-CXP, Axion-CL, Cyton-CXP Y window closed interrupt.
<b>INT_V_ACQUIRED</b>	R/W, CON489[2], Aon-CXP, Axion-CL, Cyton-CXP V window closed interrupt..
<b>INT_ENC_B</b>	R/W, CON489[4], Aon-CXP, Axion-CL, Cyton-CXP Encoder B interrupt.
<b>INT_ENC_A</b>	R/W, CON489[5], Aon-CXP, Axion-CL, Cyton-CXP Encoder A interrupt.
<b>INT_TRIG</b>	R/W, CON489[6], Aon-CXP, Axion-CL, Cyton-CXP Trigger interrupt.
<b>INT_Z_START</b>	R/W, CON489[7], Aon-CXP, Axion-CL, Cyton-CXP Start of Z Window interrupt.
<b>INT_Y_START</b>	R/W, CON489[8], Aon-CXP, Axion-CL, Cyton-CXP Start of Y Window interrupt.
<b>INT_V_START</b>	R/W, CON489[9], Aon-CXP, Axion-CL, Cyton-CXP Start of V Window interrupt.
<b>INT_BM_ERROR</b>	R/W, CON489[26], Aon-CXP, Axion-CL, Cyton-CXP Buffer manager interrupt.

<b>INT_AE_LOSS_OF_SYNC</b>	R/W, CON489[27], Aon-CXP, Axion-CL, Cyton-CXP Loss of sync in the Acquisition Engine interrupt.
<b>INT_PCIE_PKT_DROPPED</b>	R/W, CON489[28], Aon-CXP, Axion-CL, Cyton-CXP PCIe packet dropped interrupt.
<b>INT_Z_ACQUIRED_LEGACY</b>	R/W, CON489[29], Aon-CXP, Axion-CL, Cyton-CXP Copy of INT_Z_ACQUIRED.



## 2.21 CON490

<b>Bit</b>	<b>Name</b>
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, Cyton-CXP

T is at least on active interrupt on the board.

**ENINT\_ALL** R/W, CON490[8], Aon-CXP, Axion-CL, Cyton-CXP

Set to 1 to enable board interrupts.

## 2.22 CON548

Bit	Name
0	INT_Z_ACQUIRED_M
1	INT_Y_ACQUIRED_M
2	INT_V_ACQUIRED_M
3	Reserved
4	INT_ENC_B_M
5	INT_ENC_A_M
6	INT_TRIG_M
7	INT_Z_START_M
8	INT_Y_START_M
9	INT_V_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_Z_ACQUIRED_LEGACY_M
30	Reserved
31	Reserved

<b>INT_Z_ACQUIRED_M</b>	R/W, CON548[0], Aon-CXP, Axion-CL, Cyton-CXP INT_Z_ACQUIRED mask.
<b>INT_Y_ACQUIRED_M</b>	R/W, CON548[1], Aon-CXP, Axion-CL, Cyton-CXP INT_Y_ACQUIRED mask.
<b>INT_V_ACQUIRED_M</b>	R/W, CON548[2], Aon-CXP, Axion-CL, Cyton-CXP INT_V_ACQUIRED mask.
<b>INT_ENC_B_M</b>	R/W, CON548[4], Aon-CXP, Axion-CL, Cyton-CXP INT_ENC_B mask.
<b>INT_ENC_A_M</b>	R/W, CON548[5], Aon-CXP, Axion-CL, Cyton-CXP INT_ENC_A mask.
<b>INT_TRIG_M</b>	R/W, CON548[6], Aon-CXP, Axion-CL, Cyton-CXP INT_TRIG mask.
<b>INT_Z_START_M</b>	R/W, CON548[7], Aon-CXP, Axion-CL, Cyton-CXP INT_Z_START mask.
<b>INT_Y_START_M</b>	R/W, CON548[8], Aon-CXP, Axion-CL, Cyton-CXP INT_Y_START mask.
<b>INT_V_START_M</b>	R/W, CON548[9], Aon-CXP, Axion-CL, Cyton-CXP INT_V_START mask.
<b>INT_BM_ERROR_M</b>	R/W, CON548[26], Aon-CXP, Axion-CL, Cyton-CXP INT_BM_ERROR mask.

**INT\_AE\_LOSS\_OF\_SYNC\_M** R/W, CON548[27], Aon-CXP, Axion-CL, Cyton-CXP

INT\_AE\_LOSS\_OF\_SYNC mask.

**INT\_PCIE\_PKT\_DROPPED\_M** R/W, CON548[28], Aon-CXP, Axion-CL, Cyton-CXP

INT\_PCIE\_PKT\_DROPPED mask.

**INT\_Z\_ACQUIRED\_LEGACY\_M** R/W, CON548[29], Aon-CXP, Axion-CL, Cyton-CXP

INT\_Z\_ACQUIRED\_LEGACY mask.

## 2.23 CON549

Bit	Name
0	INT_Z_ACQUIRED_WP
1	INT_Y_ACQUIRED_WP
2	INT_V_ACQUIRED_WP
3	Reserved
4	INT_ENC_B_WP
5	INT_ENC_A_WP
6	INT_TRIG_WP
7	INT_Z_START_WP
8	INT_Y_START_WP
9	INT_V_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_Z_ACQUIRED_LEGACY_WP
30	Reserved
31	Reserved

<b>INT_Z_ACQUIRED_WP</b>	R/W, CON549[0], Aon-CXP, Axion-CL, Cyton-CXP INT_Z_ACQUIRED write protect.
<b>INT_Y_ACQUIRED_WP</b>	R/W, CON549[1], Aon-CXP, Axion-CL, Cyton-CXP INT_Y_ACQUIRED write protect.
<b>INT_V_ACQUIRED_WP</b>	R/W, CON549[2], Aon-CXP, Axion-CL, Cyton-CXP INT_V_ACQUIRED write protect.
<b>INT_ENC_B_WP</b>	R/W, CON549[4], Aon-CXP, Axion-CL, Cyton-CXP INT_ENC_B write protect.
<b>INT_ENC_A_WP</b>	R/W, CON549[5], Aon-CXP, Axion-CL, Cyton-CXP INT_ENC_A write protect.
<b>INT_Z_START_WP</b>	R/W, CON548[7], Aon-CXP, Axion-CL, Cyton-CXP INT_Z_START write protect.
<b>INT_Y_START_WP</b>	R/W, CON548[8], Aon-CXP, Axion-CL, Cyton-CXP INT_Y_START write protect.
<b>INT_V_START_WP</b>	R/W, CON548[9], Aon-CXP, Axion-CL, Cyton-CXP INT_V_START write protect.
<b>INT_TRIG_WP</b>	R/W, CON549[6], Aon-CXP, Axion-CL, Cyton-CXP INT_TRIG write protect.
<b>INT_BM_ERROR_WP</b>	R/W, CON549[26], Aon-CXP, Axion-CL, Cyton-CXP INT_BM_ERROR write protect.

**INT\_AE\_LOSS\_OF\_SYNC\_WP** R/W, CON549[27], Aon-CXP, Axion-CL, Cyton-CXP

INT\_AE\_LOSS\_OF\_SYNC write protect.

**INT\_PCIE\_PKT\_DROPPED\_WP** R/W, CON549[28], Aon-CXP, Axion-CL, Cyton-CXP

INT\_PCIE\_PKT\_DROPPED write protect.

**INT\_Z\_ACQUIRED\_LEGACY\_WP** R/W, CON549[29], Aon-CXP, Axion-CL, Cyton-CXP

INT\_Z\_ACQUIRED\_LEGACY write protect.



## 2.24 SF\_DIM

<b>Bit</b>	<b>Name</b>
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, 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, Cyton-CXP

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

## 2.25 SF\_CON

<b>Bit</b>	<b>Name</b>
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, 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, Cyton-CXP

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

SF_STATE	Meaning
0	Free run
1	Triggered
2	Reserved
3	Reserved

**SF\_MODE** R/W, SF\_CON[5..4], Aon-CXP, Axion-CL, Cyton-CXP

Describe SF\_MODE .

**SF\_LINE\_SCAN** R/W, SF\_CON[7], Aon-CXP, Axion-CL, 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, 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, Cyton-CXP

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

<b>SF_Y_GAP</b>	R/W, SF_CON[23..20], Aon-CXP, Axion-CL, Cyton-CXP The number of lines between frames.
<b>SF_Z_GAP</b>	R/W, SF_CON[27..24], Aon-CXP, Axion-CL, Cyton-CXP The number of frames between volumes.
<b>SF_INC_X</b>	R/W, SF_CON[28], Aon-CXP, Axion-CL, Cyton-CXP The amount to increment the grey scale output value every pixel.
<b>SF_INC_Y</b>	R/W, SF_CON[29], Aon-CXP, Axion-CL, Cyton-CXP The amount to increment the grey scale output value every line.
<b>SF_INC_Z</b>	R/W, SF_CON[30], Aon-CXP, Axion-CL, Cyton-CXP The amount to increment the grey scale output value every frame.



# 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 and the Cyton-CXP.

From a software perspective, the StreamSync system is compatible with the previous BitFlow products. However, digging deeper, these new systems 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 precaches 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 Quad 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, 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, 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

<b>Bit</b>	<b>Name</b>
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, 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, 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, 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, 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, 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

<b>Bit</b>	<b>Name</b>
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, Cyton-CXP

The current value of the on board switch SW1.

**CPLD\_MODE** RO, BOARD\_CONFIG[7..4], Aon-CXP, Axion-CL, 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, Cyton-CXP

The current value of the three on board straps.

### 3.8 PACKETS\_SENT\_STATUS

<b>Bit</b>	<b>Name</b>
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, 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, 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

<b>Bit</b>	<b>Name</b>
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, 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, 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

<b>Bit</b>	<b>Name</b>
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, 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, 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, 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, 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, Cyton-CXP

Reserved.

**RD\_ON\_EMPTY** RO, PKT\_STAT[13], Aon-CXP, Axion-CL, Cyton-CXP

FIFO underflow in PacketEngine.

**WR\_ON\_FULL** R/W, PKT\_STAT[14], Aon-CXP, Axion-CL, Cyton-CXP

FIFO overflow in PacketEngine.

**PKT\_FLUSH\_**  
**ENABLE**

R/W, PKT\_STAT[16], Aon-CXP, Axion-CL, 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 PacketEngine 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, 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, 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, Cyton-CXP

Returns the current state of the Buffer Manager.

<b>BM_STATE</b>	<b>Meaning</b>
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, Cyton-CXP

PCIe completion status from last received Quad.

**BM\_QUADS\_CACHED** RO, BUF\_MGR\_STATUS[15..8], Aon-CXP, Axion-CL, Cyton-CXP

Number of QUADS currently in the cache.

**DST\_ADDR\_ERROR\_LSB** RO, BUF\_MGR\_STATUS[20], Aon-CXP, Axion-CL, Cyton-CXP

Quad destination address is not 16 byte aligned.

**NEXT\_ADDR\_ERROR\_LSB** RO, BUF\_MGR\_STATUS[21], Aon-CXP, Axion-CL, 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, Cyton-CXP

Quad size is not a multiple of 16 bytes..

**SIZE\_ERROR\_MSB** RO, BUF\_MGR\_STATUS[23], Aon-CXP, Axion-CL, Cyton-CXP

Quad size is > 4K.

**CPL\_ERROR**

RO, BUF\_MGR\_STATUS[28], Aon-CXP, Axion-CL, 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, Cyton-CXP

Actual quad number does not match expected.

**QUAD\_FIFO\_  
OVERFLOW**

RO, BUF\_MGR\_STATUS[30], Aon-CXP, Axion-CL, Cyton-CXP

Quad cache overflowed.

**QUAD\_  
TIMEOUT\_  
DETECTED**

RO, BUF\_MGR\_STATUS[31], Aon-CXP, Axion-CL, 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

<b>Bit</b>	<b>Name</b>
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, 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, 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, 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, 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-CXP, the Axion-CL and the Cyton-CXP. 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 reproduce (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 design 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, for example, 1.2345678 seconds long. This is done by programing the TS table with three sub pulse as shown below:

- Entry 1:  $12 * 100 \text{ milliseconds} = 1.2 \text{ seconds}$
- Entry 2:  $345 * 100 \text{ microseconds} = 0.0345 \text{ seconds}$
- Entry 3:  $678 * 100 \text{ nanoseconds} = 0.0000678 \text{ seconds}$

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 pules. 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, when one trigger produces one pulse.

## 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	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
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	Reserved
25	Reserved
26	Reserved
27	Reserved
28	TS_TRIG_SEL
29	TS_TRIG_SEL
30	TS_TRIG_SEL
31	TS_TRIG_SEL

**TS\_RUN\_LEVEL** R/W, TS\_CONTROL[2..0], Aon-CXP, Axion-CL, 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, 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, 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, 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, Cyton-CXP

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

**TS\_IDX\_JUMP** R/W, TS\_CONTROL[23..16], Aon-CXP, Axion-CL, 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**

R/W, TS\_CONTROL[28..31], Aon-CXP, Axion-CL, Cyton-CXP

These bits select the source of the TS trigger.

<b>TS_TRIG_SEL</b>	<b>Meaning</b>
0 (000b)	Selected trigger (VGFx_TRIG_SEL)
1 (001b)	Selected encoder A (VFGx_ENCA_SEL)
2 (010b)	Selected encoder B (VFGx_ENCB_SEL)
3 (011b)	Selected quad encoder output (VFGx_ENCQ_SEL)
4 (100b)	Gated trigger (VGFx_TRIG_SEL gated by VFGx_ENCB_SEL)
5 (101b)	Selected encoder divider output (VFGx_ENCDIV_SEL)

### 4.3 TS\_TABLE\_CONTROL

<b>Bit</b>	<b>Name</b>
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, 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

<b>Bit</b>	<b>Name</b>
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	Reserved
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, 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, 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, Cyton-CXP

The level of the CT0 signal for this pulse.

**TS\_STATE\_CT1** R/W, TS\_TABLE\_ENTRY[13], Aon-CXP, Axion-CL, Cyton-CXP

The level of the CT1 signal for this pulse.

**TS\_STATE\_CT2** R/W, TS\_TABLE\_ENTRY[14], Aon-CXP, Axion-CL, Cyton-CXP

The level of the CT2 signal for this pulse.

**TS\_STATE\_CT3** R/W, TS\_TABLE\_ENTRY[15], Aon-CXP, Axion-CL, Cyton-CXP

The level of the CT3 signal for this pulse.

**TS\_COUNT** R/W, TS\_TABLE\_ENTRY[26..17], Aon-CXP, Axion-CL, 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, 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, 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, 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	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
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, 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, 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, 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, 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, Cyton-CXP

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

**ATS\_IDX\_JUMP** R/W, ATS\_CONTROL[23..16], Aon-CXP, Axion-CL, 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, Cyton-CXP

These bits select the source of the ATS trigger.

<b>ATS_TRIG_SEL</b>	<b>Meaning</b>
0 (000b)	Selected trigger (VGFx_TRIG_SEL)
1 (001b)	Selected encoder A (VFGx_ENCA_SEL)
2 (010b)	Selected encoder B (VFGx_ENCB_SEL)
3 (011b)	Selected quad encoder output (VFGx_ENCO_SEL)
4 (100b)	Gated trigger (VGFx_TRIG_SEL gated by VFGx_ENCB_SEL)
5 (101b)	Selected encoder divider output (VFGx_ENCDIV_SEL)

## 4.6 ATS\_TABLE\_CONTROL

<b>Bit</b>	<b>Name</b>
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, 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

<b>Bit</b>	<b>Name</b>
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	Reserved
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, 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, 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, Cyton-CXP

The level of the CT0 signal for this pulse.

**ATS\_STATE\_CT1** R/W, ATS\_TABLE\_ENTRY[13], Aon-CXP, Axion-CL, Cyton-CXP

The level of the CT1 signal for this pulse.

**ATS\_STATE\_CT2** R/W, ATS\_TABLE\_ENTRY[14], Aon-CXP, Axion-CL, Cyton-CXP

The level of the CT2 signal for this pulse.

**ATS\_STATE\_CT3** R/W, ATS\_TABLE\_ENTRY[15], Aon-CXP, Axion-CL, Cyton-CXP

The level of the CT3 signal for this pulse.

**ATS\_COUNT** R/W, ATS\_TABLE\_ENTRY[26..17], Aon-CXP, Axion-CL, 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, 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, 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, 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 And Cyton I/O System

---

## Chapter 5

### 5.1 Introduction

The I/O system on the Aon, Axion and Cyton families of frame grabbers are based on the Karbon famil 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

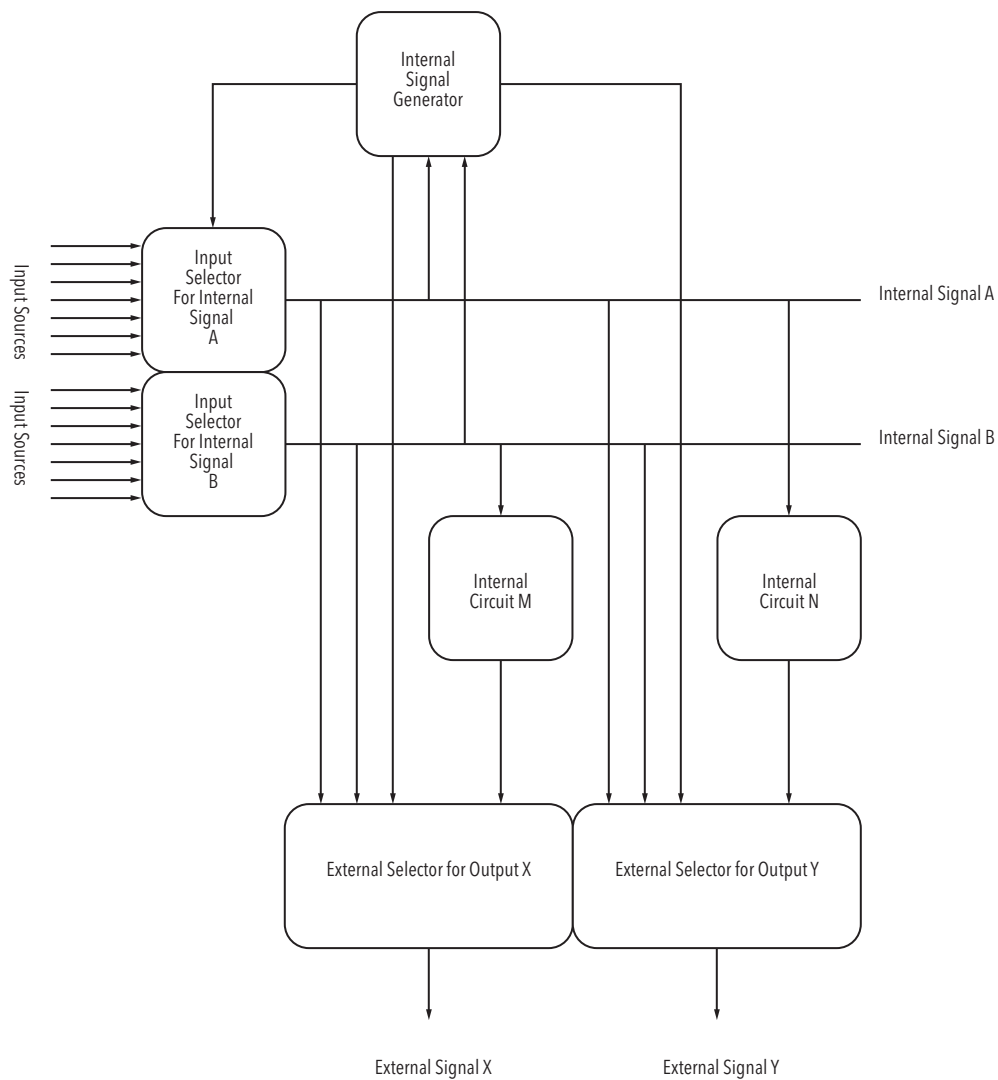
Because BitFlow's frame grabber can acquire from more than one camera, it has always been a contention between the desire to let each camera be independent, for example, each with its own trigger, and for them to be synchronized, i.e. all cameras using one trigger. The Cyton and Axion I/O system provides for 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 Aon, Axtion and Cyton 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 and the Axion-CL 1xE have only one VFG.*

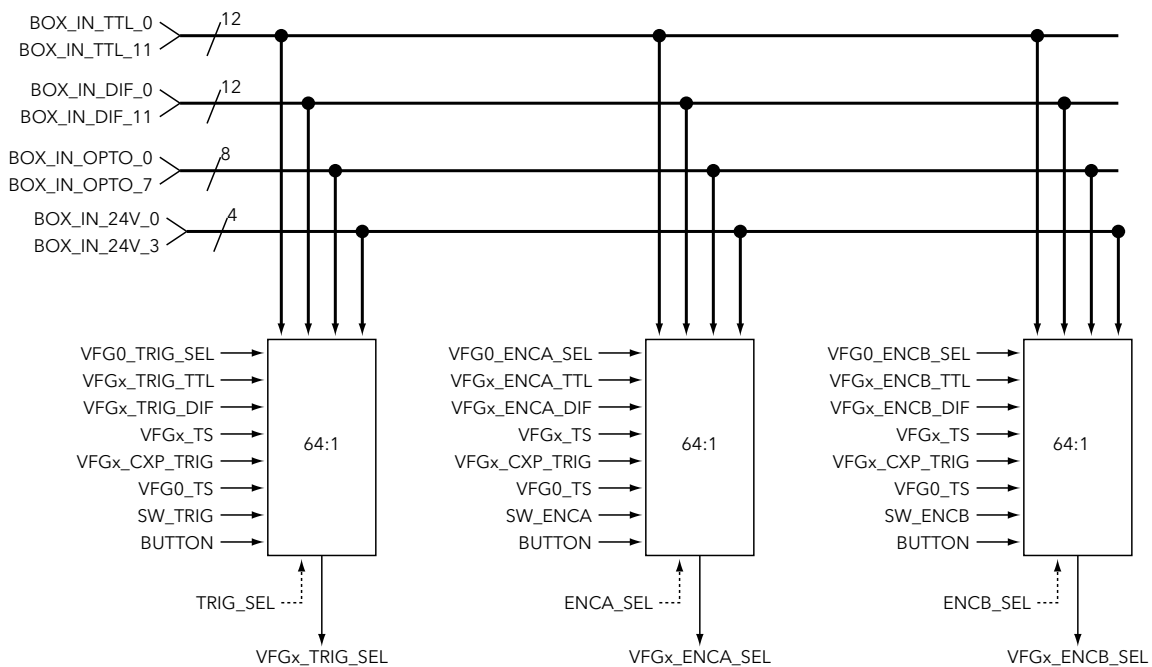


Figure 5-2 I/O System Input Selection

## 5.4 Internal Signals

There are five internal I/O signals. The “x” in the signal name refers to VFG number of the current VFG being used. In general, the “x” is always used as all VFGs are symmetrical. The one exception 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.

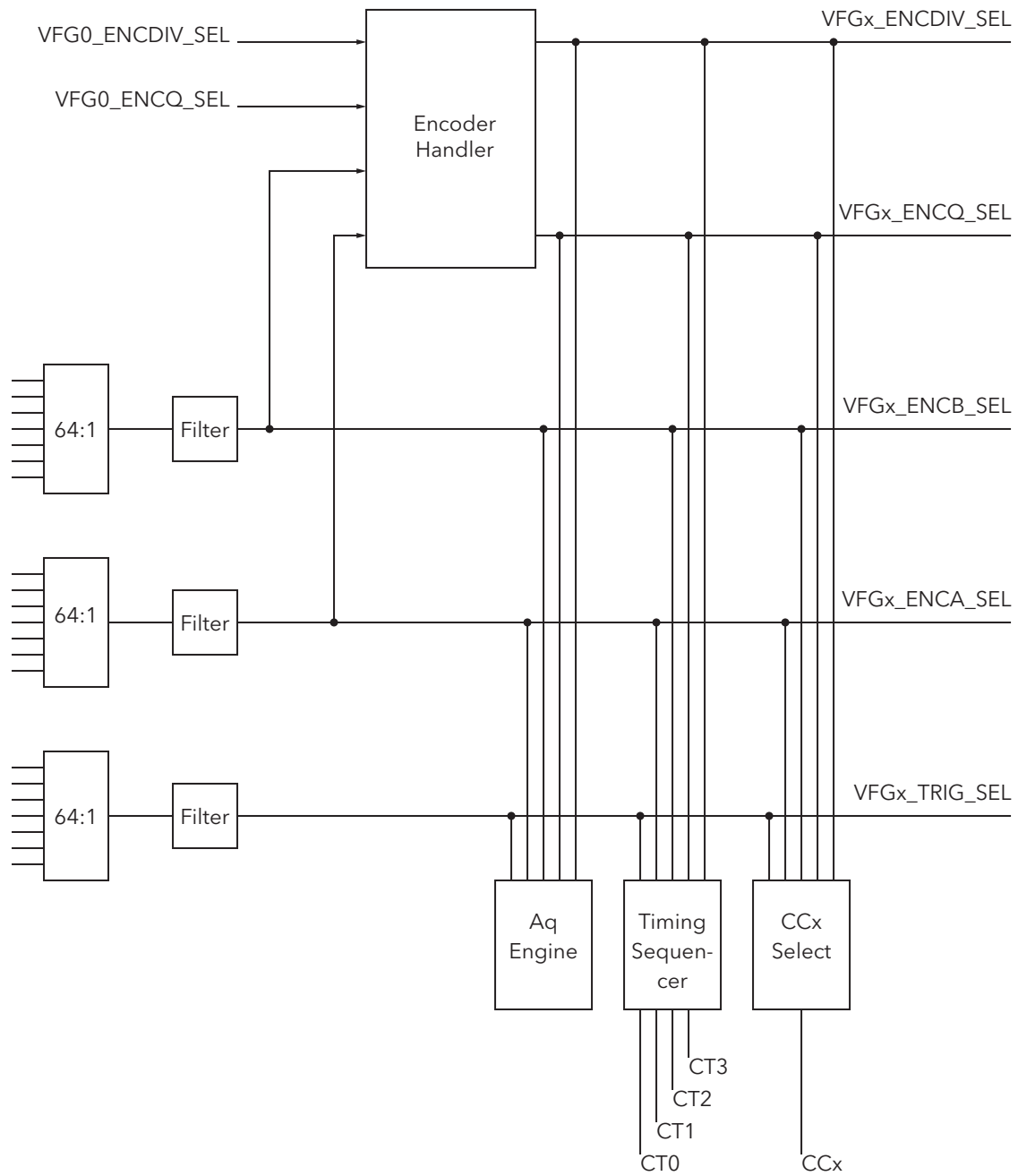


Figure 5-3 Internal Signal Routing

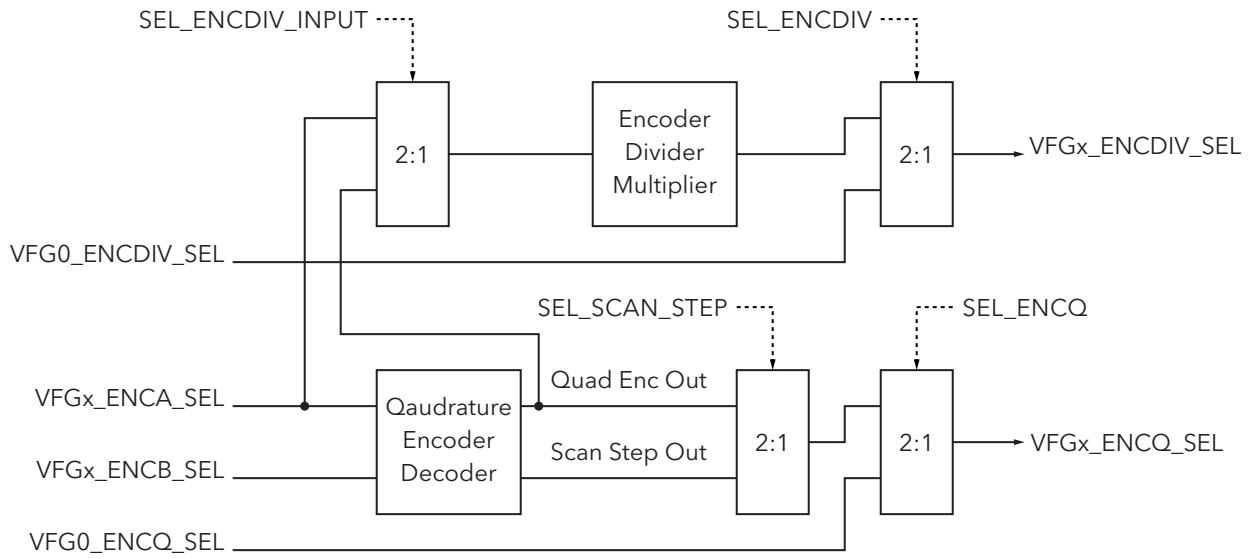


Figure 5-4 Encoder Handler

## 5.5 Output Signal Selection

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

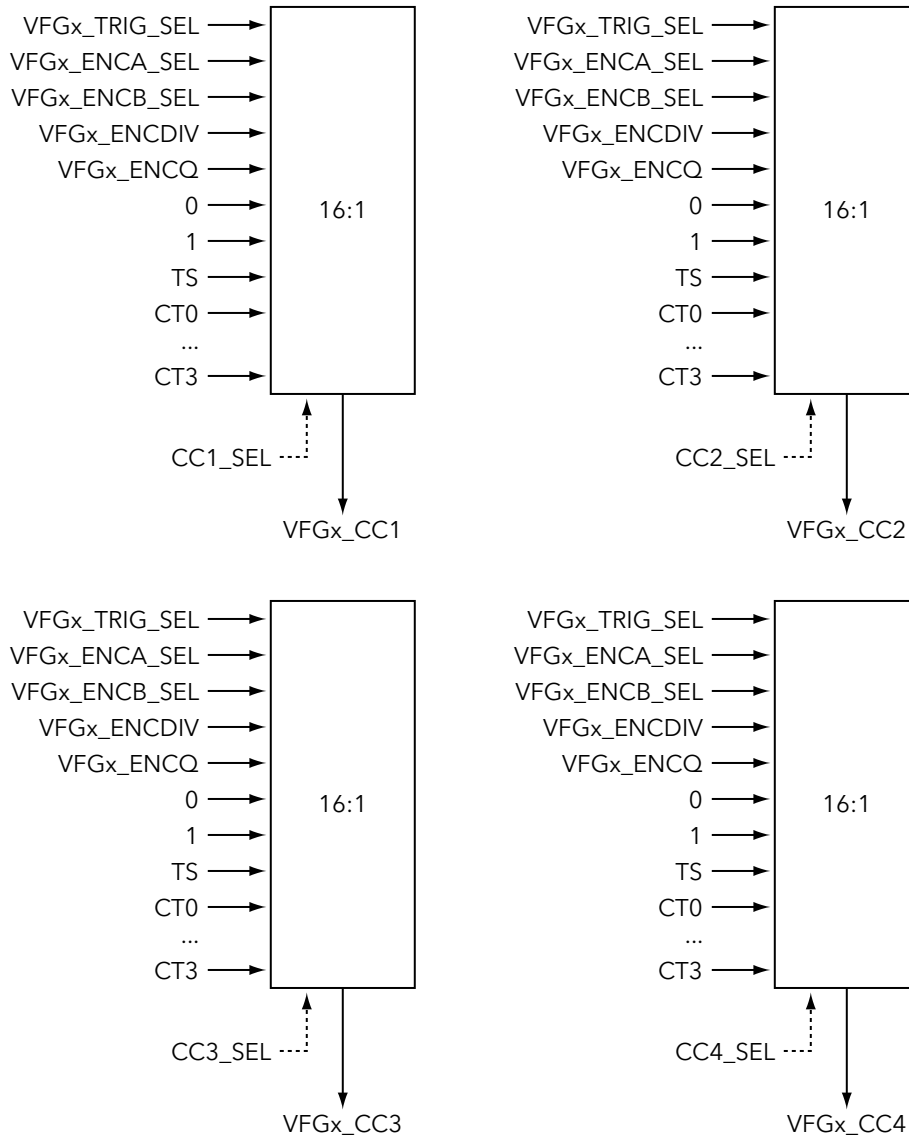


Figure 5-5 Output Signal Source Selection

## 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.*

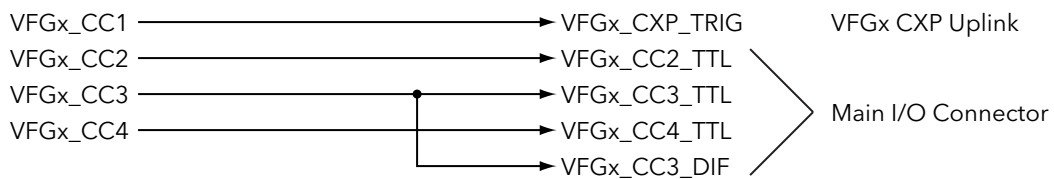


Figure 5-6 Output Signal Routing

## 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 required on the board in order to control 36 outputs.*

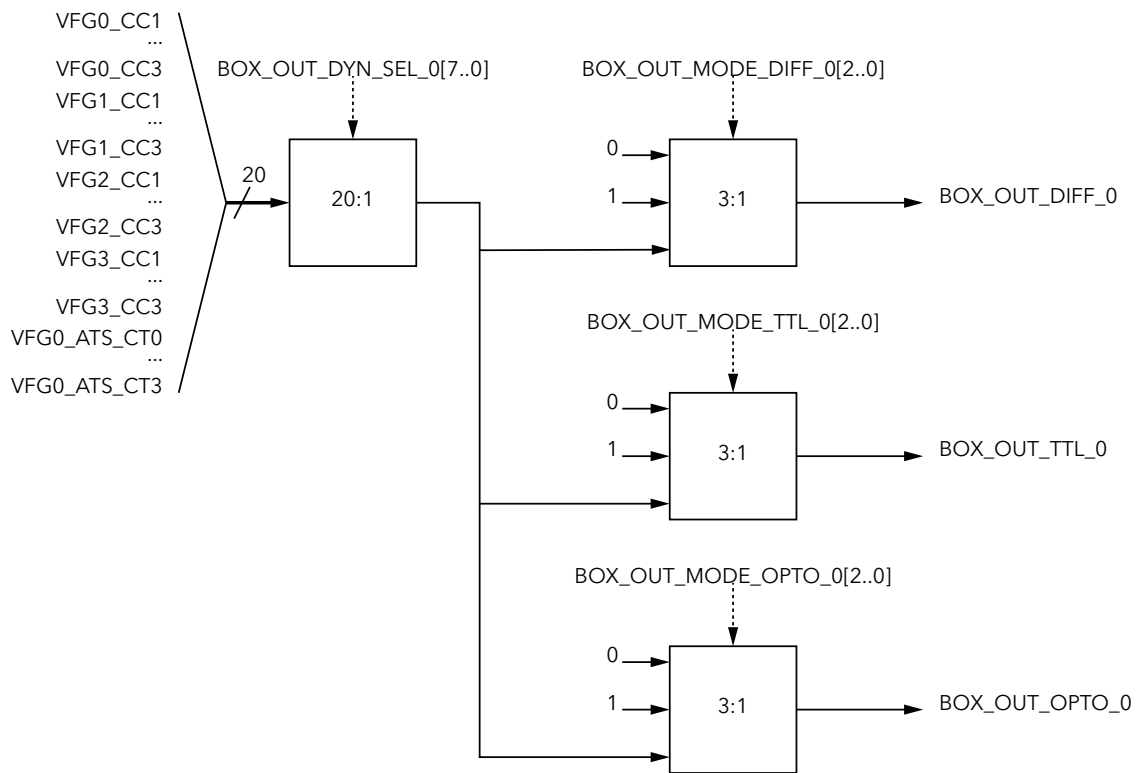


Figure 5-7 BitBox Output Signal Routing

*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 through a small cable. Contact BitFlow for more information on the BitBox.*





# The Aon, Axion and Cyton 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-CXP, Axion-CL and Cyton-CXP.

## 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

<b>RD_BOX_IN_TTL_X</b>	RO, CON60[11..0], Aon-CXP, Axion-CL, Cyton-CXP  These bits reflect the real-time state of the 12 TTL inputs on the BitBox.
<b>RD_BOX_IN_DIF_X</b>	RO, CON60[23..12], Aon-CXP, Axion-CL, Cyton-CXP  These bits reflect the real-time state of the 12 differential inputs on the BitBox.
<b>ENINT_CXP</b>	R/W, CON60[24], Aon-CXP, Axion-CL, Cyton-CXP  This bit enables interrupts from the CXP subsystem.
<b>INT_CXP</b>	RO, CON60[25], Aon-CXP, Axion-CL, 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.
<b>SW_TRIG</b>	R/W, CON60[29], Aon-CXP, Axion-CL, 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.
<b>SW_ENCA</b>	R/W, CON60[30], Aon-CXP, Axion-CL, 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.
<b>SW_ENCB</b>	R/W, CON60[31], Aon-CXP, Axion-CL, 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

<b>RD_BOX_IN_OPTO_X</b>	RO, CON61[7..0], Aon-CXP, Axion-CL, Cyton-CXP These bits reflect the real-time state of the eight Opto-Isolated inputs on the BitBox.
<b>RD_BOX_IN_24V_X</b>	RO, CON61[11..8], Aon-CXP, Axion-CL, Cyton-CXP These bits reflect the real-time state of the four 24V inputs on the BitBox.
<b>RD_CXP_TRIG_OUT</b>	RO, CON61[23], Aon-CXP, Axion-CL, Cyton-CXP This bit reflects the real-time state of the CXP trigger signal going to the camera.

## 6.4 CON62

<b>Bit</b>	<b>Name</b>
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

<b>RD_TRIG_TTL</b>	RO, CON62[0], Aon-CXP, Axion-CL, Cyton-CXP  This bit reflects the real-time state of the VFG's TTL trigger input.
<b>RD_TRIG_DIF</b>	RO, CON62[1], Aon-CXP, Axion-CL, Cyton-CXP  This bit reflects the real-time state of the VFG's differential trigger input.
<b>RD_TRIG_VFG0</b>	RO, CON62[2], Aon-CXP, Axion-CL, Cyton-CXP  This bit reflects the real-time state of VFG0's selected trigger signal.
<b>RD_SCAN_STEP</b>	RO, CON62[3], Aon-CXP, Axion-CL, Cyton-CXP  This bit reflects the real-time state of the VFG's scan step circuitry output (from the quadrature encoder circuit).
<b>RD_SW_TRIG</b>	RO, CON62[4], Aon-CXP, Axion-CL, Cyton-CXP  This bit reflects the real-time state of the VFG's software trigger.
<b>RD_ENCA_TTL</b>	RO, CON62[5], Aon-CXP, Axion-CL, Cyton-CXP  This bit reflects the real-time state of the VFG's TTL encoder A input.
<b>RD_ENCA_DIF</b>	RO, CON62[6], Aon-CXP, Axion-CL, Cyton-CXP  This bit reflects the real-time state of the VFG's differential encoder A input.
<b>RD_ENCA_VFG0</b>	RO, CON62[7], Aon-CXP, Axion-CL, Cyton-CXP  This bit reflects the real-time state of VFG0's selected encoder A signal.
<b>RD_ENCA_SW</b>	RO, CON62[8], Aon-CXP, Axion-CL, Cyton-CXP  This bit reflects the real-time state of the VFG's software encoder A.
<b>RD_ENCB_TTL</b>	RO, CON62[9], Aon-CXP, Axion-CL, Cyton-CXP  This bit reflects the real-time state of the VFG's TTL encoder B input.

<b>RD_ENCB_DIF</b>	RO, CON62[10], Aon-CXP, Axion-CL, Cyton-CXP
	This bit reflects the real-time state of the VFG's differential encoder B input.
<b>RD_ENCB_VFG0</b>	RO, CON62[11], Aon-CXP, Axion-CL, Cyton-CXP
	This bit reflects the real-time state of VFG0's selected encoder B signal.
<b>RD_ENCB_SW</b>	RO, CON62[12], Aon-CXP, Axion-CL, Cyton-CXP
	This bit reflects the real-time state of the VFG's software encoder B.
<b>RD_BUTTON</b>	RO, CON62[13], Aon-CXP, Axion-CL, Cyton-CXP
	This bit reflects the real-time state of the VFG's button input.
<b>DIV_RESET_DISABLE</b>	R/W, CON62[15], Aon-CXP, Axion-CL, Cyton-CXP
	This bit disables the Divider/Multiplier auto-reset function.
<b>RD_CXP_TRIG_IN</b>	RO, CON62[24], Aon-CXP, Axion-CL, Cyton-CXP
	This bit reflects the real-time state of the CXP trigger signal coming from the camera.
<b>EN_TRIG</b>	R/W, CON62[25], Aon-CXP, Axion-CL, Cyton-CXP
	This bit is used to enable the selected trigger.
<b>EN_ENCA</b>	R/W, CON62[26], Aon-CXP, Axion-CL, Cyton-CXP
	This bit is used to enable the selected encoder A.
<b>EN_ENCB</b>	R/W, CON62[27], Aon-CXP, Axion-CL, Cyton-CXP
	This bit is used to enable the selected encoder B.
<b>RD_ENCB_SELECTED</b>	RO, CON62[29], Aon-CXP, Axion-CL, 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, 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, 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, Cyton-CXP

Selects the VFG's trigger source.

<b>SEL_TRIG</b>	<b>Source</b>
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, Cyton-CXP

Selects this VFG's encoder A source.

<b>SEL_ENCA</b>	<b>Source</b>
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

<b>SEL_ENCA</b>	<b>Source</b>
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, Cyton-CXP

Selects the source of encoder B.

<b>SEL_ENCB</b>	<b>Source</b>
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, Cyton-CXP

Selects the source of CC1.

<b>SEL_CC1</b>	<b>Source</b>
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

**SEL\_CC2** R/W, CON63[25..22], Aon-CXP, Axion-CL, Cyton-CXP

Selects the source of CC2.

<b>SEL_CC2</b>	<b>Source</b>
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)
6 (0110b)	VFGx_TRIG_SEL
7 (0111b)	VFGx_ENCA_SEL
8 (1000b)	VFGx_ENCB_SEL

<b>SEL_CC2</b>	<b>Source</b>
9 (1001b)	VFG0_CT0
10 (1010b)	VFG0_CT1
11 (1011b)	VFG0_CT2
12 (1100b)	VFG0_CT3
13 (1101b)	VFGx_ENCDIV_SEL
14 (1110b)	VFGx_ENCO_SEL

**SEL\_LED**

R/W, CON63[31..28], Aon-CXP, Axion-CL, 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.

<b>SEL_CC1</b>	<b>Source</b>
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
12 (1100b)	VFG0_NTG or VFG0_TS
13 (1101b)	AQSTAT[1]
14 (1110b)	Overstep, OVS
15 (1111b)	Reserved

## 6.6 CON64

<b>Bit</b>	<b>Name</b>
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	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	LED_RED
29	LED_ORANGE
30	LED_GREEN
31	LED_BLUE

**SEL\_CC3**

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

Selects the source of CC3.

<b>SEL_CC3</b>	<b>Source</b>
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_ENCO_SEL

**SEL\_CC4**

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

Selects the source of CC4.

<b>SEL_CC4</b>	<b>Source</b>
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



<b>SEL_CC4</b>	<b>Source</b>
9 (1001b)	VFG0_CT0
10 (1010b)	VFG0_CT1
11 (1011b)	VFG0_CT2
12 (1100b)	VFG0_CT3
13 (1101b)	VFGx_ENCDIV_SEL
14 (1110b)	VFGx_ENCO_SEL

**TRIGPOL**

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

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

<b>TRIGPOL</b>	<b>Meaning</b>
0	Trigger asserted on rising edge
1	Trigger asserted on falling edge

**ENCA\_POL**

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

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

<b>ENCA_POL</b>	<b>Meaning</b>
0	Encoder A asserted on rising edge
1	Encoder A asserted on falling edge

**ENCB\_POL**

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

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

<b>ENCB_POL</b>	<b>Meaning</b>
0	Encoder B asserted on rising edge
1	Encoder B asserted on falling edge

**LED\_RED**

R/W, CON64[28], Aon-CXP, Axion-CL, 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, 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, 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 ADDR\_TRIG\_FILTER

<b>Bit</b>	<b>Name</b>
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, ADDR\_TRIG\_FILTER[17..0], Aon-CXP, Axion-CL, 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 noised and will be filtered out. Any pulses over this size will be consider signal. The units of this register are 4 nano-seconds. 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 ADDR\_ENCA\_FILTER

<b>Bit</b>	<b>Name</b>
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, ADDR\_ENCA\_FILTER[17..0], Aon-CXP, Axion-CL, 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 noised 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 ADDR\_ENCB\_FILTER

<b>Bit</b>	<b>Name</b>
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, ADDR\_ENCB\_FILTER[17..0], Aon-CXP, Axion-CL, 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 noised 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

<b>Bit</b>	<b>Name</b>
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, 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: Sources that start with VFG1, VFG2 and VFG3 are not available on the Axion-1xE. Sources that start with VFG3 are not available on the Cyton-CXP2 or the Axion-2xE.*

**BOX\_OUT\_DYN\_SEL\_1** R/W, BOX\_OUT\_DYN\_SEL\_SET\_A[15..8], Aon-CXP, Axion-CL, 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, 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, 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, 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, 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, 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, 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

<b>Bit</b>	<b>Name</b>
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, 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, 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, Cyton-CXP

This register controls the dynamic source for the three BitBox output signals BOX\_OUT\_DIFF\_10, BOX\_OUT\_TTL\_10 and BOX\_OUT\_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, 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, 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, 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, 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, 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, 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_4	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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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\_9 bitfield.

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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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 the 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 ration numbers in the available scaling range can be selected (t 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 incoming encoder rate into the camera's line rate:

$$F_{\text{out}} = F_{\text{in}} \frac{2^N}{M}$$

W:

$F_{\text{out}}$  = The frequency used to driver the camera or the NTG or the CTags

$F_{\text{in}}$  = 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 ( $F_{\text{in}}$ ) is 10 KHz and that we need an output ( $F_{\text{out}}$ ) 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, tfore, 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 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 frequency. In this mode the output will track the input using the following formula (the variables are the same as in Section 7.2.1)

$$F_{\text{out}} = \frac{F_{\text{in}}}{4M}$$

### DC Mode (ENC\_DIV\_FORCE\_DC = 1)

In this mode, the  $F_{\text{in}}$  goes below 1.6 KHz,  $F_{\text{out}}$  will go 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  $F_{\text{in}}$  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**

<b>Name</b>	<b>Purpose</b>
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 input 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 register is the heart of the encoder system. For example, trigger 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 only 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 acquire 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 if the stage is moving in one direction only. 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, it 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, where an image is needed in steps across the range of the stage. This mode is enabled 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 8-1 are all the registers that can be used.:

**Table 8-1 Observability Registers.**

<b>Register</b>	<b>Meaning</b>
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**

<b>Encoder</b>	<b>Frame Grabber</b>
A	VFGx_ENCODERA_TTL
B	VFGx_ENCODERB_TTL
Ground	GND

**Table 8-3 LVDS Quadrature Encoder Connections**

<b>Encoder</b>	<b>Frame Grabber</b>
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.

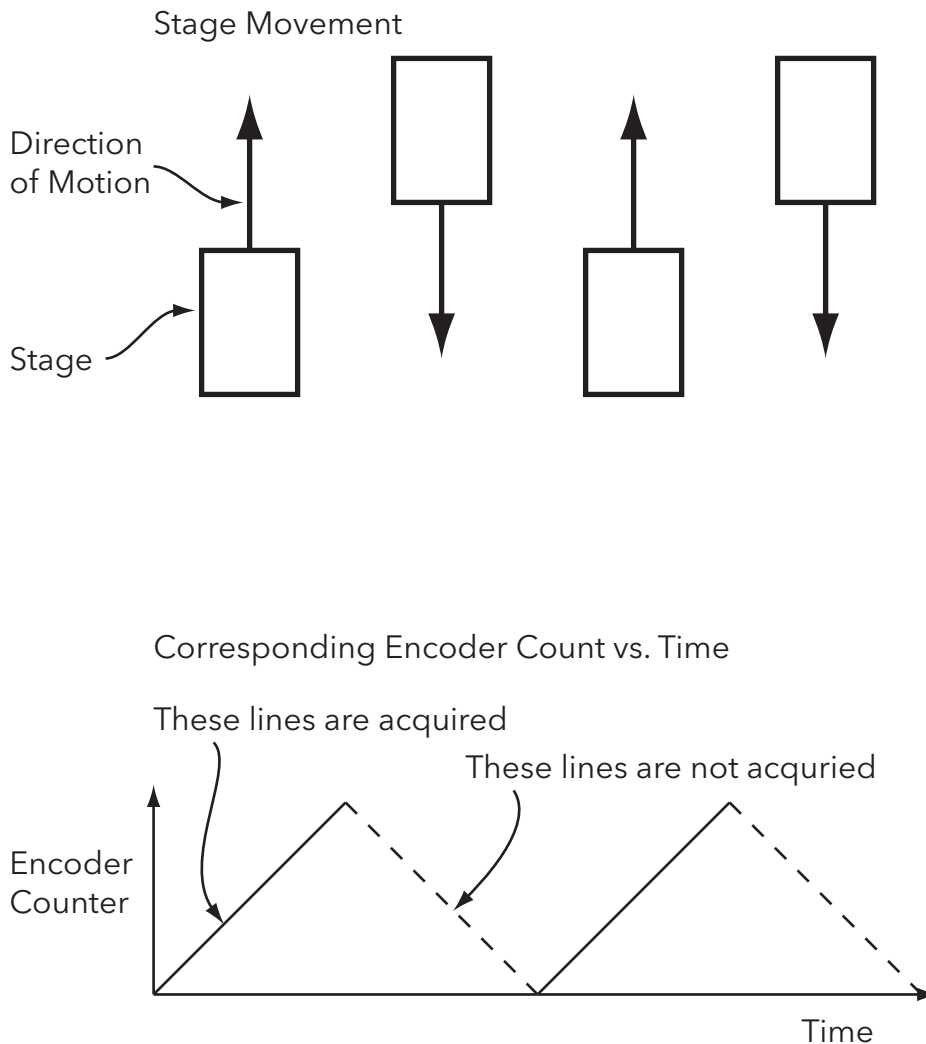


Figure 8-1 Encoder Count vs Time

Figure 8-2 shows all of the major quadrature encoder modes.

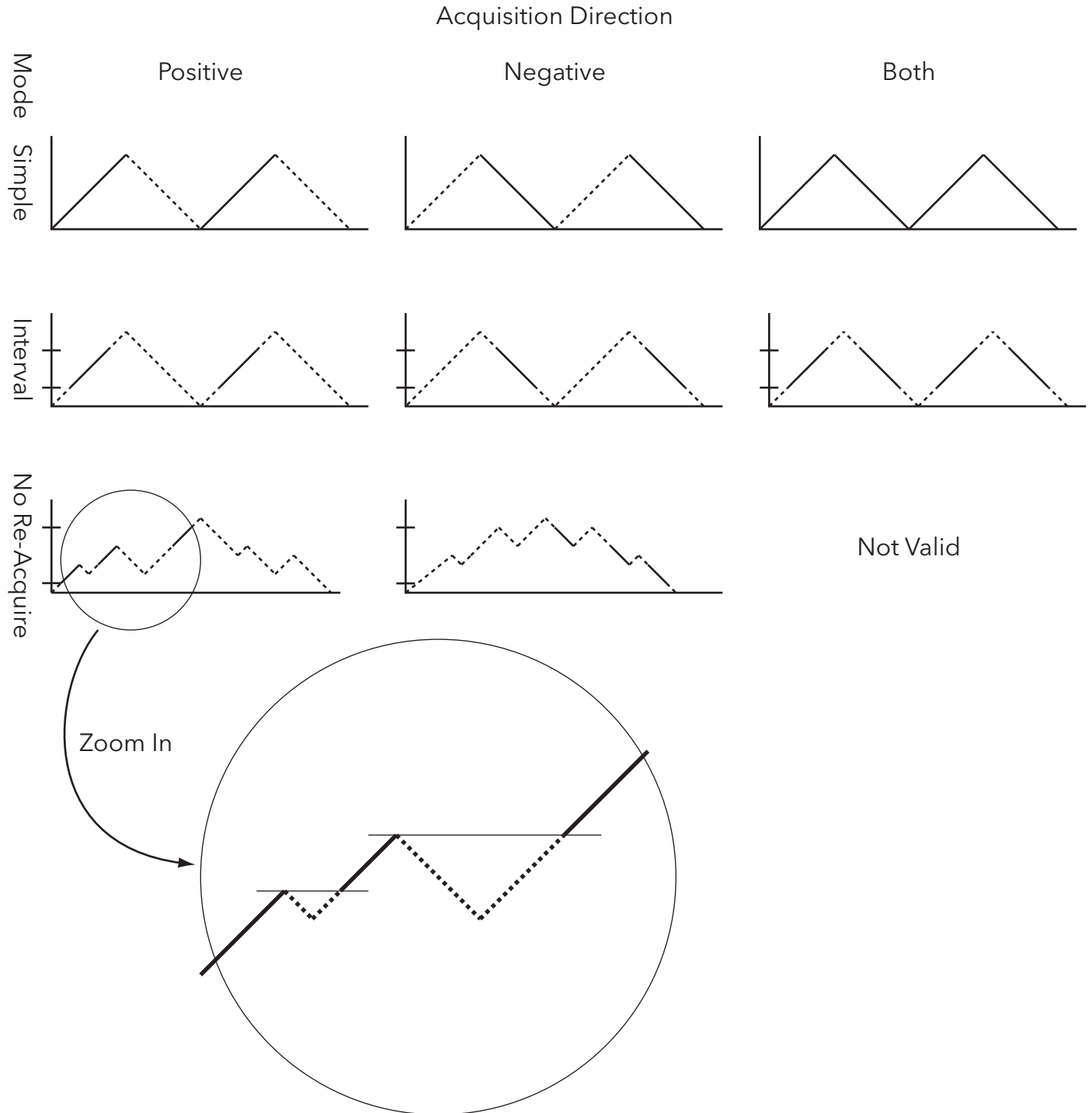


Figure 8-2 Quadrature Encoder Modes vs. Acquisition





# 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\_ENCO** R/W, CON65[0], Aon-CXP, Axion-CL, Cyton-CXP

This bit selects which quadrature encoder circuit output will be used on this VFG.

SEL_ENCO	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, Cyton-CXP

This bit selects which input will driver the encoder divider circuit.

SEL_EENC_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, 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 output
1	Select the output of VFG0's encoder divider output

**ENC\_DIV\_N** R/W, CON65[5..3], Aon-CXP, Axion-CL, 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, 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, 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

<b>Bit</b>	<b>Name</b>
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	QENC_DUAL_PHASE
30	SCAN_STEP_TRIG
31	QENC_RESET

**QENC\_INTRVL\_LL** R/W/R/W, CON66[23..0], Aon-CXP, Axion-CL, 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, 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" modes.
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, 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, 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 QENC\_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, 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 t 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)

### QENC\_DUAL\_PHASE

R/W, CON66[29], Aon-CXP, Axion-CL, Cyton-CXP

This bit controls which type of encoder is attached.

QENC_DUAL_PHASE	Meaning
0	A single phase encoder is attached
1	A quadrature encoder is attached

### SCAN\_STEP\_TRIG

R/W, CON66[30], Aon-CXP, Axion-CL, 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	Trigger comes of the normal source
1	Trigger comes from the scan step circuit

**QENC\_RESET**      WO, CON66[31], Aon-CXP, Axion-CL, 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, 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\_REAQ\_MODE** R/W, CON67[25..24], Aon-CXP, Axion-CL, 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, there 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, 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 where 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, 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, 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, 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

<b>Bit</b>	<b>Name</b>
0	RD_ENCO_SELECTED
1	RD_ENCDIV_SELECTED
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

**RD\_ENCO\_  
SELECTED**

RO, CON68[0], Aon-CXP, Axion-CL, Cyton-CXP

This bit indicates the current state of selected quad encoder circuit output.

**RD\_ENCDIV\_  
SELECTED**

RO, CON68[1], Aon-CXP, Axion-CL, Cyton-CXP

This bit displays the current state of the selected encoder divider output.

**ENC\_DIV\_RESET**

WO, CON68[2], Aon-CXP, Axion-CL, 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, 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, Cyton-CXP

This bitfield displays the current quadrature encoder count.

**QENC\_PHASEA** RO, CON69[24], Aon-CXP, Axion-CL, Cyton-CXP

This bit displays the current logic level of the A quadrature encoder phase.

**QENC\_PHASEB** RO, CON69[25], Aon-CXP, Axion-CL, Cyton-CXP

This bit displays the current logic level of the B quadrature encoder phase.

**QENC\_DIR** RO, CON69[26], Aon-CXP, Axion-CL, 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, 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, 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 if 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, 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-CXP and the Cyton-CXP. Most of these registers are programmed automatically by the software and are probably only need be accessed under very special circumstances.

The Cyton-CXP has up to four CXP links the Aon-CXP has only one link. Each link has it's 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**

<b>Register Example Name</b>	<b>CXP Link number</b>
0_POCPX_EN_POWER	0
1_POCPX_EN_POWER	1
2_POCPX_EN_POWER	2
3_POCPX_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 camera, in which case each VFG gets date 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 one or 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_POCPX_EN_POWER
1	0_POCPX_EN_24V_REG
2	0_POCPX_EN_CAM_SENSE
3	0_POCPX_CAM_IS_POCPX
4	0_POCPX_SHORT_DETECTED
5	0_POCPX_OPEN_DETECTED
6	0_POCPX_OVER_DETECTED
7	0_POCPX_OVER_LATCH
8	0_POCPX_UNDER_DETECTED
9	0_POCPX_UNDER_LATCH
10	0_POCPX_24V_OK
11	Reserved
12	0_POCPX_STATE
13	0_POCPX_OVR_AUTO_RESTART
14	0_POCPX_SENSE_BYPASS
15	0_ENABLE_POCPX_SYSTEM
16	0_POCPX_CURRENT_LATCH
17	0_POCPX_CURRENT_LATCH
18	0_POCPX_CURRENT_LATCH
19	0_POCPX_CURRENT_LATCH
20	0_POCPX_CURRENT_LATCH
21	0_POCPX_CURRENT_LATCH
22	0_POCPX_CURRENT_LATCH
23	0_POCPX_CURRENT_LATCH
24	0_POCPX_CURRENT
25	0_POCPX_CURRENT
26	0_POCPX_CURRENT
27	0_POCPX_CURRENT
28	0_POCPX_CURRENT
29	0_POCPX_CURRENT
30	0_POCPX_CURRENT
31	0_POCPX_CURRENT

<b>0_POCPX_EN_POWER</b>	RO, CON104[0], Aon-CXP, Cyton-CXP CXP Power enabled.
<b>0_POCPX_EN_24V_REG</b>	RO, CON104[1], Aon-CXP, Cyton-CXP 24V Regulator enabled.
<b>0_POCPX_EN_CAM_SENSE</b>	RO, CON104[2], Aon-CXP, Cyton-CXP Enable POCXP Sense.
<b>0_POCPX_CAM_IS_POCPX</b>	RO, CON104[3], Aon-CXP, Cyton-CXP Reserved for future use.
<b>0_POCPX_SHORT_DETECTED</b>	RO, CON104[4], Aon-CXP, Cyton-CXP Only valid if POCXP_EN_CAM_SENSE is asserted. This status bit tells if an short circuit was detected from the camera in response to POCXP_EN_CAM_SENSE.
<b>0_POCPX_OPEN_DETECTED</b>	RO, CON104[5], Aon-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.
<b>0_POCPX_OVER_DETECTED</b>	RO, CON104[6], Aon-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.
<b>0_POCPX_OVER_LATCH</b>	RO, CON104[7], Aon-CXP, Cyton-CXP Latched version of POCXP_OVER_DETECTED. Cleared by hardware when both POCXP_EN_POWER and POCXP_EN_24V_REG transition from 0 to 1.

<b>0_POCPX_UNDER_DETECTED</b>	RO, CON104[8], Aon-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.
<b>0_POCPX_UNDER_LATCH</b>	RO, CON104[9], Aon-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.
<b>0_POCPX_24V_OK</b>	RO, CON104[10], Aon-CXP, Cyton-CXP  TBD
<b>0_POCPX_STATE</b>	RO, CON104[12], Aon-CXP, Cyton-CXP  TBD
<b>0_POCPX_OVR_AUTO_RESTART</b>	RW, CON104[13], Aon-CXP, Cyton-CXP  TBD
<b>0_POCPX_SENSE_BYPASS</b>	RW, CON104[14], Aon-CXP, Cyton-CXP  TBD
<b>0_ENABLE_POCPX_SYSTEM</b>	RW, CON104[15], Aon-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.
<b>0_POCPX_CURRENT_LATCH</b>	RO, CON104[23..16], Aon-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\_POCPX\_  
CURRENT**

RO, CON104[31..24], Aon-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

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

**0\_POCPX\_  
OVER\_TIMER**

R/W, CON105[31..0], Aon-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

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



**0\_POCPX\_  
UNDER\_TIMER**

R/W, CON106[31..0], Aon-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, 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, Cyton-CXP

Depth of the control channel request fifo. It is measured in 32-bit words.

## 10.6 CON108

<b>Bit</b>	<b>Name</b>
0	0_COM_SEND_FIFO_CLR
1	0_COM_SEND_GO
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_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, Cyton-CXP

Clear the control channel request fifo.

**0\_COM\_SEND\_  
GO**

WO, CON108[1], Aon-CXP, 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\_SEND\_  
FIFO\_CNT**

RO, CON108[31..16], Aon-CXP, Cyton-CXP

Number of 32-bit words currently in the request fifo.

## 10.7 CON109

<b>Bit</b>	<b>Name</b>
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, 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, Cyton-CXP

Clear the control channel receive fifo.

**0\_COM\_RCV\_  
FIFO\_CNT**

RO, CON110[31..16], Aon-CXP, Cyton-CXP

Number of 32-bit words currently in the response fifo.

## 10.9 CON111

<b>Bit</b>	<b>Name</b>
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, Cyton-CXP

Read the head of the control channel response fifo. Fifo level can be monitored with COM\_RCV\_FIFO\_CNT.

## 10.10 CON115

<b>Bit</b>	<b>Name</b>
0	O_LINK_INT_DEST
1	O_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

**O\_LINK\_INT\_DEST** R/W, CON115[1..0], Aon-CXP, Cyton-CXP  
TBD

## 10.11 CON116

<b>Bit</b>	<b>Name</b>
0	0_UNDER_CURRENT
1	0_OVER_CURRENT
2	0_TRIG_ACK_RCVD
3	0_GPIO_ACK_RCVD
4	0_CTL_ACK_RCVD
5	0_GPIO_RCVD
6	0_TRIG_RCVD
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_IOACK_UNEXPECTED_INT
14	0_IOACK_NOMATCH2
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

<b>0_UNDER_CURRENT</b>	R/W, CON116[0], Aon-CXP, Cyton-CXP Undercurrent detected by POCXP controller interrupt.
<b>0_OVER_CURRENT</b>	R/W, CON116[1], Aon-CXP, Cyton-CXP Overcurrent detected by POCXP controller interrupt.
<b>0_TRIG_ACK_RCVD</b>	R/W, CON116[2], Aon-CXP, Cyton-CXP Trigger acknowledgment received from device interrupt.
<b>0_GPIO_ACK_RCVD</b>	R/W, CON116[3], Aon-CXP, Cyton-CXP GPIO acknowledgment received from device interrupt.
<b>0_CTL_ACK_RCVD</b>	R/W, CON116[4], Aon-CXP, Cyton-CXP Control packet acknowledgment received from device interrupt.
<b>0_GPIO_RCVD</b>	R/W, CON116[5], Aon-CXP, Cyton-CXP GPIO received from device interrupt.
<b>0_TRIG_RCVD</b>	R/W, CON116[6], Aon-CXP, Cyton-CXP Trigger received from device interrupt.
<b>0_CTL_RSP_FIFO_OVF</b>	R/W, CON116[7], Aon-CXP, Cyton-CXP Overflow detected in the control response fifo (Device to host direction) interrupt.
<b>0_CTL_REQ_FIFO_OVF</b>	R/W, CON116[8], Aon-CXP, Cyton-CXP Overflow detected in control request fifo (Host to device direction) interrupt.
<b>0_DOWN_TRIG_RCVD</b>	R/W, CON116[9], Aon-CXP, Cyton-CXP Received CXP trigger from camera.

<b>0_TRIG_NOMATCH</b>	R/W, CON116[10], Aon-CXP, Cyton-CXP Problem decoding a trigger packet from device interrupt.
<b>0_IOACK_UNKNOWN_TYPE</b>	R/W, CON116[11], Aon-CXP, Cyton-CXP Problem decoding an IO acknowledgment from device interrupt.
<b>0_IOACK_NOMATCH</b>	R/W, CON116[12], Aon-CXP, Cyton-CXP Problem decoding an IO acknowledgment from device interrupt.
<b>0_IOACK_UNEXPECTED_INT</b>	R/W, CON116[13], Aon-CXP, Cyton-CXP Problem decoding an IO acknowledgment from device interrupt.
<b>0_IOACK_NOMATCH2</b>	R/W, CON116[14], Aon-CXP, Cyton-CXP Problem decoding an IO acknowledgment from device interrupt.
<b>0_STRM_PKT_DROP</b>	R/W, CON116[15], Aon-CXP, Cyton-CXP Problem decoding a stream packet header, remainder of packet is dropped interrupt.
<b>0_STRM_NOT_ENOUGH_DAT</b>	R/W, CON116[16], Aon-CXP, Cyton-CXP Not implemented, reserved interrupt.
<b>0_STRM_TOO_MUCH_DAT</b>	R/W, CON116[17], Aon-CXP, Cyton-CXP Not implemented, reserved interrupt.
<b>0_STRM_BAD_CRC</b>	R/W, CON116[18], Aon-CXP, Cyton-CXP CRC error detected in stream packet interrupt.
<b>0_STRM_OVERFLOW</b>	R/W, CON116[19], Aon-CXP, Cyton-CXP Overflow in the stream packet buffer interrupt.



**0\_STRM\_  
CORNER**

R/W, CON116[19], Aon-CXP, Cyton-CXP

Not implemented, reserved interrupt.

**0\_SERDES\_  
LOST\_ALIGN**

R/W, CON116[21], Aon-CXP, 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

<b>0_UNDER_CURRENT_M</b>	R/W, CON117[0], Aon-CXP, Cyton-CXP Mask for the 0_UNDER_CURRENT_M interrupt.
<b>0_OVER_CURRENT_M</b>	R/W, CON117[1], Aon-CXP, Cyton-CXP Mask for the 0_OVER_CURRENT_M interrupt.
<b>0_TRIG_ACK_RCVD_M</b>	R/W, CON117[2], Aon-CXP, Cyton-CXP Mask for the 0_TRIG_ACK_RCVD_M interrupt.
<b>0_GPIO_ACK_RCVD_M</b>	R/W, CON117[3], Aon-CXP, Cyton-CXP Mask for the 0_GPIO_ACK_RCVD_M interrupt.
<b>0_CTL_ACK_RCVD_M</b>	R/W, CON117[4], Aon-CXP, Cyton-CXP Mask for the 0_CTL_ACK_RCVD_M interrupt.
<b>0_GPIO_RCVD_M</b>	R/W, CON117[5], Aon-CXP, Cyton-CXP Mask for the 0_GPIO_RCVD_M interrupt.
<b>0_TRIG_RCVD_M</b>	R/W, CON117[6], Aon-CXP, Cyton-CXP Mask for the 0_TRIG_RCVD_M interrupt.
<b>0_CTL_RSP_FIFO_OVF_M</b>	R/W, CON117[7], Aon-CXP, Cyton-CXP Mask for the 0_CTL_RSP_FIFO_OVF_M interrupt.
<b>0_CTL_REQ_FIFO_OVF_M</b>	R/W, CON117[8], Aon-CXP, Cyton-CXP Mask for the 0_CTL_REQ_FIFO_OVF_M interrupt.
<b>0_DOWN_TRIG_RCVD_M</b>	R/W, CON117[9], Aon-CXP, Cyton-CXP Mask for the 0_DOWN_TRIG_RCVD_M interrupt.

<b>0_TRIG_NOMATCH_M</b>	R/W, CON117[10], Aon-CXP, Cyton-CXP Mask for the 0_TRIG_NOMATCH_M interrupt.
<b>0_IOACK_UNKNOWN_TYPE_M</b>	R/W, CON117[11], Aon-CXP, Cyton-CXP Mask for the 0_IOACK_UNKNOWN_TYPE_M interrupt.
<b>0_IOACK_NOMATCH_M</b>	R/W, CON117[12], Aon-CXP, Cyton-CXP Mask for the 0_IOACK_NOMATCH_M interrupt.
<b>0_IOACK_UNEXPECTED_INT_M</b>	R/W, CON117[13], Aon-CXP, Cyton-CXP Mask for the 0_IOACK_UNEXPECTED_INT_M interrupt.
<b>0_IOACK_NOMATCH2_M</b>	R/W, CON117[14], Aon-CXP, Cyton-CXP Mask for the 0_IOACK_NOMATCH2_M interrupt.
<b>0_STRM_PKT_DROP_M</b>	R/W, CON117[15], Aon-CXP, Cyton-CXP Mask for the 0_STRM_PKT_DROP_M interrupt.
<b>0_STRM_NOT_ENOUGH_DAT_M</b>	R/W, CON117[16], Aon-CXP, Cyton-CXP Mask for the 0_STRM_NOT_ENOUGH_DAT_M interrupt.
<b>0_STRM_TOO_MUCH_DAT_M</b>	R/W, CON117[17], Aon-CXP, Cyton-CXP Mask for the 0_STRM_TOO_MUCH_DAT_Mv
<b>0_STRM_BAD_CRC_M</b>	R/W, CON117[18], Aon-CXP, Cyton-CXP Mask for the 0_STRM_BAD_CRC_M interrupt.
<b>0_STRM_OVERFLOW_M</b>	R/W, CON117[19], Aon-CXP, 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

<b>0_UNDER_CURRENT_WP</b>	R/W, CON118[0], Aon-CXP, Cyton-CXP Write mask for the 0_UNDER_CURRENT_WP interrupt.
<b>0_OVER_CURRENT_WP</b>	R/W, CON118[1], Aon-CXP, Cyton-CXP Write mask for the 0_OVER_CURRENT_WP interrupt.
<b>0_TRIG_ACK_RCVD_WP</b>	R/W, CON118[2], Aon-CXP, Cyton-CXP Write mask for the 0_TRIG_ACK_RCVD_WP interrupt.
<b>0_GPIO_ACK_RCVD_WP</b>	R/W, CON118[3], Aon-CXP, Cyton-CXP Write mask for the 0_GPIO_ACK_RCVD_WP interrupt.
<b>0_CTL_ACK_RCVD_WP</b>	R/W, CON118[4], Aon-CXP, Cyton-CXP Write mask for the 0_CTL_ACK_RCVD_WP interrupt.
<b>0_GPIO_RCVD_WP</b>	R/W, CON118[5], Aon-CXP, Cyton-CXP Write mask for the 0_GPIO_RCVD_WP interrupt.
<b>0_TRIG_RCVD_WP</b>	R/W, CON118[6], Aon-CXP, Cyton-CXP Write mask for the 0_TRIG_RCVD_WP interrupt.
<b>0_CTL_RSP_FIFO_OVF_WP</b>	R/W, CON118[7], Aon-CXP, Cyton-CXP Write mask for the 0_CTL_RSP_FIFO_OVF_WP interrupt.
<b>0_CTL_REQ_FIFO_OVF_WP</b>	R/W, CON118[8], Aon-CXP, Cyton-CXP Write mask for the 0_CTL_REQ_FIFO_OVF_WP interrupt.
<b>0_DOWN_TRIG_RCVD_WP</b>	R/W, CON118[9], Aon-CXP, Cyton-CXP Write mask for the 0_DOWN_TRIG_RCVD_WP interrupt.

<b>0_TRIG_NOMATCH_WP</b>	R/W, CON118[10], Aon-CXP, Cyton-CXP Write mask for the 0_TRIG_NOMATCH_WP interrupt.
<b>0_IOACK_UNKNOWN_TYPE_WP</b>	R/W, CON118[11], Aon-CXP, Cyton-CXP Write mask for the 0_IOACK_UNKNOWN_TYPE_WP interrupt.
<b>0_IOACK_NOMATCH_WP</b>	R/W, CON118[12], Aon-CXP, Cyton-CXP Write mask for the 0_IOACK_NOMATCH_WP interrupt.
<b>0_IOACK_UNEXPECTED_INT_WP</b>	R/W, CON118[13], Aon-CXP, Cyton-CXP Write mask for the 0_IOACK_UNEXPECTED_INT_WP interrupt.
<b>0_IOACK_NOMATCH2_WP</b>	R/W, CON118[14], Aon-CXP, Cyton-CXP Write mask for the 0_IOACK_NOMATCH2_WP interrupt.
<b>0_STRM_PKT_DROP_WP</b>	R/W, CON118[15], Aon-CXP, Cyton-CXP Write mask for the 0_STRM_PKT_DROP_WP interrupt.
<b>0_STRM_NOT_ENOUGH_DAT_WP</b>	R/W, CON118[16], Aon-CXP, Cyton-CXP Write mask for the 0_STRM_NOT_ENOUGH_DAT_WP interrupt.
<b>0_STRM_TOO_WPUCH_DAT_WP</b>	R/W, CON118[17], Aon-CXP, Cyton-CXP Write mask for the 0_STRM_TOO_WPUCH_DAT_WP interrupt.
<b>0_STRM_BAD_CRC_WP</b>	R/W, CON118[18], Aon-CXP, Cyton-CXP Write mask for the 0_STRM_BAD_CRC_WP interrupt.
<b>0_STRM_OVERFLOW_WP</b>	R/W, CON118[19], Aon-CXP, Cyton-CXP Write mask for the 0_STRM_OVERFLOW_WP interrupt.

## 10.14 CON120

<b>Bit</b>	<b>Name</b>
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, Cyton-CXP

Number of CXP packets received on this link. The entire register [31:0] is cleared on a write access of any value.

**0\_PKT\_GNT\_  
CNT**

RO, CON120[31..16], Aon-CXP, Cyton-CXP

Number of packets forwarded from this link to the Stream Assembler engine. The entire register [31:0] is cleared on a write access of any value.

## 10.15 CON121

<b>Bit</b>	<b>Name</b>
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, 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, 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

<b>0_CXP_TRIG_FALL_SENT</b>	RO, CON122[7..0], Aon-CXP, Cyton-CXP Number of CXP trigger falling edge trigger packets sent to the camera.
<b>0_CXP_TRIG_RISE_SENT</b>	RO, CON122[15..8], Aon-CXP, Cyton-CXP Number of CXP trigger rising edge trigger packets sent to the camera.
<b>0_CXP_TRIG_ACK_CNT</b>	RO, CON122[23..16], Aon-CXP, Cyton-CXP Number of CXP trigger acknowledgments received from the camera.
<b>0_CXP_TRIG_STATE</b>	RO, CON122[31], Aon-CXP, 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	Reserved
27	Reserved
28	Reserved
29	Reserved
30	0_DT_CLR_CNT
31	0_DT_STATE

<b>0_DT_FALL_RCVD_COUNT</b>	RO, CON123[7..0], Aon-CXP, Cyton-CXP Number of CXP down stream (camera to frame grabber) trigger falling edges.
<b>0_DT_RISE_RCVD_COUNT</b>	RO, CON123[15..8], Aon-CXP, Cyton-CXP Number of CXP down stream (camera to frame grabber) trigger rising edges.
<b>0_DT_TRIG_DELAY</b>	RO, CON123[23..16], Aon-CXP, Cyton-CXP The payload of the CXP down stream (camera to frame grabber) trigger.
<b>0_DT_FALL_DISABLE</b>	R/W, CON123[24], Aon-CXP, 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.
<b>0_DT_RISE_DISABLE</b>	R/W, CON123[25], Aon-CXP, 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.
<b>0_DT_CLR_CNT</b>	WO, CON123[30], Aon-CXP, Cyton-CXP Writing this register to 1 will clear that counters in this registers.
<b>0_DT_STATE</b>	RO, CON123[31], Aon-CXP, Cyton-CXP Current stated of the downstream trigger.

## 10.18 CON127

<b>Bit</b>	<b>Name</b>
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



<b>0_SERDES_STATE</b>	RO, CON127[7..0], Aon-CXP, Cyton-CXP Current state of the SERDES associated with this link.
<b>0_LINK_SPEED</b>	RO, CON127[15..8], Aon-CXP, Cyton-CXP Current speed of the SERDES associated with this link.
<b>0_LOST_ALIGN_CNT</b>	RO, CON127[25..16], Aon-CXP, Cyton-CXP Number of times the SERDES associated with this link has lost alignment.
<b>0_RESYNC_TOG_MSTR</b>	RO, CON127[28], Aon-CXP, Cyton-CXP Diagnostics.
<b>0_RESYNC_TOG</b>	RO, CON127[29], Aon-CXP, Cyton-CXP Diagnostics.
<b>0_SERDES_ALIGNED</b>	RO, CON127[30], Aon-CXP, Cyton-CXP If this bit is 1 the SERDES associated with this link is aligned.

## 10.19 CON128

<b>Bit</b>	<b>Name</b>
0	O_RAW_DATA_MODE
1	O_REMOVE_IDLE
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\_RAW\_DATA\_  
MODE**

R/W, CON128[0], Aon-CXP, 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, Cyton-CXP

Remove CXP idle packets from the raw capture.

## 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, Cyton-CXP

Describe 0\_SERDES\_ERROR\_CODE

## 10.21 CON136

Bit	Name
0	1_POEXX_EN_POWER
1	1_POEXX_EN_24V_REG
2	1_POEXX_EN_CAM_SENSE
3	1_POEXX_CAM_IS_POEXX
4	1_POEXX_SHORT_DETECTED
5	1_POEXX_OPEN_DETECTED
6	1_POEXX_OVER_DETECTED
7	1_POEXX_OVER_LATCH
8	1_POEXX_UNDER_DETECTED
9	1_POEXX_UNDER_LATCH
10	1_POEXX_24V_OK
11	Reserved
12	1_POEXX_STATE
13	1_POEXX_OVR_AUTO_RESTART
14	1_POEXX_SENSE_BYPASS
15	1_ENABLE_POEXX_SYSTEM
16	1_POEXX_CURRENT_LATCH
17	1_POEXX_CURRENT_LATCH
18	1_POEXX_CURRENT_LATCH
19	1_POEXX_CURRENT_LATCH
20	1_POEXX_CURRENT_LATCH
21	1_POEXX_CURRENT_LATCH
22	1_POEXX_CURRENT_LATCH
23	1_POEXX_CURRENT_LATCH
24	1_POEXX_CURRENT
25	1_POEXX_CURRENT
26	1_POEXX_CURRENT
27	1_POEXX_CURRENT
28	1_POEXX_CURRENT
29	1_POEXX_CURRENT
30	1_POEXX_CURRENT
31	1_POEXX_CURRENT

<b>1_POCPX_EN_POWER</b>	RO, CON136[0], Cyton-CXP See description of 0_POCPX_EN_POWER
<b>1_POCPX_EN_24V_REG</b>	RO, CON136[1], Cyton-CXP See description of 0_POCPX_EN_24V_REG
<b>1_POCPX_EN_CAM_SENSE</b>	RO, CON136[2], Cyton-CXP See description of 0_POCPX_EN_CAM_SENSE
<b>1_POCPX_CAM_IS_POCPX</b>	RO, CON136[3], Cyton-CXP See description of 0_POCPX_CAM_IS_POCPX
<b>1_POCPX_SHORT_DETECTED</b>	RO, CON136[4], Cyton-CXP See description of 0_POCPX_SHORT_DETECTED
<b>1_POCPX_OPEN_DETECTED</b>	RO, CON136[5], Cyton-CXP See description of 0_POCPX_OPEN_DETECTED
<b>1_POCPX_OVER_DETECTED</b>	RO, CON136[6], Cyton-CXP See description of 0_POCPX_OVER_DETECTED
<b>1_POCPX_OVER_LATCH</b>	RO, CON136[7], Cyton-CXP See description of 0_POCPX_OVER_LATCH
<b>1_POCPX_UNDER_DETECTED</b>	RO, CON136[8], Cyton-CXP See description of 0_POCPX_UNDER_DETECTED
<b>1_POCPX_UNDER_LATCH</b>	RO, CON136[9], Cyton-CXP See description of 0_POCPX_UNDER_LATCH

<b>1_POCPX_24V_OK</b>	RO, CON136[10], Cyton-CXP See description of 0_POCPX_24V_OK
<b>1_POCPX_STATE</b>	RO, CON136[12], Cyton-CXP See description of 0_POCPX_STATE.
<b>1_POCPX_OVR_AUTO_RESTART</b>	RW, CON136[13], Cyton-CXP See description of 0_POCPX_OVR_AUTO_RESTART.
<b>1_POCPX_SENSE_BYPASS</b>	RW, CON136[14], Cyton-CXP See description of 0_POCPX_SENSE_BYPASS.
<b>1_ENABLE_POCPX_SYSTEM</b>	RW, CON136[15], Cyton-CXP See description of 0_ENABLE_POCPX_SYSTEM.
<b>1_POCPX_CURRENT_LATCH</b>	RO, CON136[23..16], Cyton-CXP See description of 0_POCPX_CURRENT_LATCH
<b>1_POCPX_CURRENT</b>	RO, CON136[31..24], Cyton-CXP See description of 0_POCPX_CURRENT



## 10.22 CON137

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

**1\_POCPX\_  
OVER\_TIMER**

R/W, CON137[31..0], Cyton-CXP

See description of 0\_POCPX\_OVER\_TIMER

## 10.23 CON138

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

**1\_POCPX\_  
UNDER\_TIMER**

R/W, CON138[31..0], Cyton-CXP

See description of 0\_POCPX\_UNDER\_TIMER

## 10.24 CON139

<b>Bit</b>	<b>Name</b>
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], Cyton-CXP  
See description of 0\_COM\_RCV\_FIFO\_SIZE

**1\_COM\_SEND\_FIFO\_SIZE** RO, CON139[31..16], 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	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_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], Cyton-CXP  
See description of 0\_COM\_SEND\_FIFO\_CLR

**1\_COM\_SEND\_GO** WO, CON140[1], Cyton-CXP  
See description of 0\_COM\_SEND\_GO

**1\_COM\_SEND\_FIFO\_CNT** RO, CON140[31..16], Cyton-CXP  
See description of 0\_COM\_SEND\_FIFO\_CNT



## 10.26 CON141

<b>Bit</b>	<b>Name</b>
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], 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], Cyton-CXP

See description of 0\_COM\_RCV\_FIFO\_CLR

**1\_COM\_RCV\_  
FIFO\_CNT**

RO, CON142[31..16], Cyton-CXP

See description of 0\_COM\_RCV\_FIFO\_CNT

## 10.28 CON143

<b>Bit</b>	<b>Name</b>
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], Cyton-CXP

See description of 0\_COM\_RCV\_DATA

## 10.29 CON147

<b>Bit</b>	<b>Name</b>
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], 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_GPIO_ACK_RCVD
4	1_CTL_ACK_RCVD
5	1_GPIO_RCVD
6	1_TRIG_RCVD
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_IOACK_UNEXPECTED_INT
14	1_IOACK_NOMATCH2
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

<b>1_UNDER_CURRENT</b>	R/W, CON148[0], Cyton-CXP See description of 0_UNDER_CURRENT
<b>1_OVER_CURRENT</b>	R/W, CON148[1], Cyton-CXP See description of 0_OVER_CURRENT
<b>1_TRIG_ACK_RCVD</b>	R/W, CON148[2], Cyton-CXP See description of 0_TRIG_ACK_RCVD
<b>1_GPIO_ACK_RCVD</b>	R/W, CON148[3], Cyton-CXP See description of 0_GPIO_ACK_RCVD
<b>1_CTL_ACK_RCVD</b>	R/W, CON148[4], Cyton-CXP See description of 0_CTL_ACK_RCVD
<b>1_GPIO_RCVD</b>	R/W, CON148[5], Cyton-CXP See description of 0_GPIO_RCVD
<b>1_TRIG_RCVD</b>	R/W, CON148[6], Cyton-CXP See description of 0_TRIG_RCVD
<b>1_CTL_RSP_FIFO_OVF</b>	R/W, CON148[7], Cyton-CXP See description of 0_CTL_RSP_FIFO_OVF
<b>1_CTL_REQ_FIFO_OVF</b>	R/W, CON148[8], Cyton-CXP See description of 0_CTL_REQ_FIFO_OVF
<b>1_DOWN_TRIG_RCVD</b>	R/W, CON148[9], Cyton-CXP See description of 0_DOWN_TRIG_RCVD

<b>1_TRIG_NOMATCH</b>	R/W, CON148[10], Cyton-CXP See description of 0_TRIG_NOMATCH
<b>1_IOACK_UNKNOWN_TYPE</b>	R/W, CON148[11], Cyton-CXP See description of 0_IOACK_UNKNOWN_TYPE
<b>1_IOACK_NOMATCH</b>	R/W, CON148[12], Cyton-CXP See description of 0_IOACK_NOMATCH
<b>1_IOACK_UNEXPECTED_INT</b>	R/W, CON148[13], Cyton-CXP See description of 0_IOACK_UNEXPECTED_INT
<b>1_IOACK_NOMATCH2</b>	R/W, CON148[14], Cyton-CXP See description of 0_IOACK_NOMATCH2
<b>1_STRM_PKT_DROP</b>	R/W, CON148[15], Cyton-CXP See description of 0_STRM_PKT_DROP
<b>1_STRM_NOT_ENOUGH_DAT</b>	R/W, CON148[16], Cyton-CXP See description of 0_STRM_NOT_ENOUGH_DAT
<b>1_STRM_TOO_MUCH_DAT</b>	R/W, CON148[17], Cyton-CXP See description of 0_STRM_TOO_MUCH_DAT
<b>1_STRM_BAD_CRC</b>	R/W, CON148[18], Cyton-CXP See description of 0_STRM_BAD_CRC
<b>1_STRM_OVERFLOW</b>	R/W, CON148[19], Cyton-CXP See description of 0_STRM_OVERFLOW

## 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

<b>1_UNDER_CURRENT_M</b>	R/W, CON149[0], Cyton-CXP See description of 0_UNDER_CURRENT_M
<b>1_OVER_CURRENT_M</b>	R/W, CON149[1], Cyton-CXP See description of 0_OVER_CURRENT_M
<b>1_TRIG_ACK_RCVD_M</b>	R/W, CON149[2], Cyton-CXP See description of 0_TRIG_ACK_RCVD_M
<b>1_GPIO_ACK_RCVD_M</b>	R/W, CON149[3], Cyton-CXP See description of 0_GPIO_ACK_RCVD_M
<b>1_CTL_ACK_RCVD_M</b>	R/W, CON149[4], Cyton-CXP See description of 0_CTL_ACK_RCVD_M
<b>1_GPIO_RCVD_M</b>	R/W, CON149[5], Cyton-CXP See description of 0_GPIO_RCVD_M
<b>1_TRIG_RCVD_M</b>	R/W, CON149[6], Cyton-CXP See description of 0_TRIG_RCVD_M
<b>1_CTL_RSP_FIFO_OVF_M</b>	R/W, CON149[7], Cyton-CXP See description of 0_CTL_RSP_FIFO_OVF_M
<b>1_CTL_REQ_FIFO_OVF_M</b>	R/W, CON149[8], Cyton-CXP See description of 0_CTL_REQ_FIFO_OVF_M
<b>1_DOWN_TRIG_RCVD_M</b>	R/W, CON149[9], Cyton-CXP See description of 0_DOWN_TRIG_RCVD_M

<b>1_TRIG_NOMATCH_M</b>	R/W, CON149[10], Cyton-CXP See description of 0_TRIG_NOMATCH_M
<b>1_IOACK_UNKNOWN_TYPE_M</b>	R/W, CON149[11], Cyton-CXP See description of 0_IOACK_UNKNOWN_TYPE_M
<b>1_IOACK_NOMATCH_M</b>	R/W, CON149[12], Cyton-CXP See description of 0_IOACK_NOMATCH_M
<b>1_IOACK_UNEXPECTED_INT_M</b>	R/W, CON149[13], Cyton-CXP See description of 0_IOACK_UNEXPECTED_INT_M
<b>1_IOACK_NOMATCH2_M</b>	R/W, CON149[14], Cyton-CXP See description of 0_IOACK_NOMATCH2_M
<b>1_STRM_PKT_DROP_M</b>	R/W, CON149[15], Cyton-CXP See description of 0_STRM_PKT_DROP_M
<b>1_STRM_NOT_ENOUGH_DAT_M</b>	R/W, CON149[16], Cyton-CXP See description of 0_STRM_NOT_ENOUGH_DAT_M
<b>1_STRM_TOO_MUCH_DAT_M</b>	R/W, CON149[17], Cyton-CXP See description of 0_STRM_TOO_MUCH_DAT_M
<b>1_STRM_BAD_CRC_M</b>	R/W, CON149[18], Cyton-CXP See description of 0_STRM_BAD_CRC_M
<b>1_STRM_OVERFLOW_M</b>	R/W, CON149[19], 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

<b>1_UNDER_CURRENT_WP</b>	R/W, CON150[0], Cyton-CXP See description of 0_UNDER_CURRENT_WP
<b>1_OVER_CURRENT_WP</b>	R/W, CON150[1], Cyton-CXP See description of 0_OVER_CURRENT_WP
<b>1_TRIG_ACK_RCVD_WP</b>	R/W, CON150[2], Cyton-CXP See description of 0_TRIG_ACK_RCVD_WP
<b>1_GPIO_ACK_RCVD_WP</b>	R/W, CON150[3], Cyton-CXP See description of 0_GPIO_ACK_RCVD_WP
<b>1_CTL_ACK_RCVD_WP</b>	R/W, CON150[4], Cyton-CXP See description of 0_CTL_ACK_RCVD_WP
<b>1_GPIO_RCVD_WP</b>	R/W, CON150[5], Cyton-CXP See description of 0_GPIO_RCVD_WP
<b>1_TRIG_RCVD_WP</b>	R/W, CON150[6], Cyton-CXP See description of 0_TRIG_RCVD_WP
<b>1_CTL_RSP_FIFO_OVF_WP</b>	R/W, CON150[7], Cyton-CXP See description of 0_CTL_RSP_FIFO_OVF_WP
<b>1_CTL_REQ_FIFO_OVF_WP</b>	R/W, CON150[8], Cyton-CXP See description of 0_CTL_REQ_FIFO_OVF_WP
<b>1_DOWN_TRIG_RCVD_WP</b>	R/W, CON150[9], Cyton-CXP See description of 0_DOWN_TRIG_RCVD_WP



<b>1_TRIG_NOMATCH_WP</b>	R/W, CON150[10], Cyton-CXP See description of 0_TRIG_NOMATCH_WP
<b>1_IOACK_UNKNOWN_TYPE_WP</b>	R/W, CON150[11], Cyton-CXP See description of 0_IOACK_UNKNOWN_TYPE_WP
<b>1_IOACK_NOMATCH_WP</b>	R/W, CON150[12], Cyton-CXP See description of 0_IOACK_NOMATCH_WP
<b>1_IOACK_UNEXPECTED_INT_WP</b>	R/W, CON150[13], Cyton-CXP See description of 0_IOACK_UNEXPECTED_INT_WP
<b>1_IOACK_NOMATCH2_WP</b>	R/W, CON150[14], Cyton-CXP See description of 0_IOACK_NOMATCH2_WP
<b>1_STRM_PKT_DROP_WP</b>	R/W, CON150[15], Cyton-CXP See description of 0_STRM_PKT_DROP_WP
<b>1_STRM_NOT_ENOUGH_DAT_WP</b>	R/W, CON150[16], Cyton-CXP See description of 0_STRM_NOT_ENOUGH_DAT_WP
<b>1_STRM_TOO_WPUCH_DAT_WP</b>	R/W, CON150[17], Cyton-CXP See description of 0_STRM_TOO_WPUCH_DAT_WP
<b>1_STRM_BAD_CRC_WP</b>	R/W, CON150[18], Cyton-CXP See description of 0_STRM_BAD_CRC_WP
<b>1_STRM_OVERFLOW_WP</b>	R/W, CON150[19], Cyton-CXP See description of 0_STRM_OVERFLOW_WP

## 10.33 CON152

<b>Bit</b>	<b>Name</b>
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], Cyton-CXP  
See description of 0\_PKT\_RCVD\_CNT

**1\_PKT\_GNT\_CNT** RO, CON152[31..16], Cyton-CXP  
See description of 0\_PKT\_GNT\_CNT

## 10.34 CON153

<b>Bit</b>	<b>Name</b>
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

<b>1_PKT_DROP_CNT</b>	RO, CON153[15..0], Cyton-CXP See description of 0_PKT_DROP_CNT
<b>1_CRC_ERR_CNT</b>	RO, CON153[31..16], 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

<b>1_CXP_TRIG_FALL_SENT</b>	RO, CON154[7..0], Cyton-CXP Number of CXP trigger falling edge trigger packets sent to the camera.
<b>1_CXP_TRIG_RISE_SENT</b>	RO, CON154[15..8], Cyton-CXP Number of CXP trigger rising edge trigger packets sent to the camera.
<b>1_CXP_TRIG_ACK_CNT</b>	RO, CON154[23..16], Cyton-CXP Number of CXP trigger acknowledgments received from the camera.
<b>1_CXP_TRIG_STATE</b>	RO, CON154[31], 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	Reserved
27	Reserved
28	Reserved
29	Reserved
30	1_DT_CLR_CNT
31	1_DT_STATE



<b>1_DT_FALL_RCVD_COUNT</b>	RO, CON123[7..0], Aon-CXP, Cyton-CXP Number of CXP down stream (camera to frame grabber) trigger falling edges.
<b>1_DT_RISE_RCVD_COUNT</b>	RO, CON123[15..8], Aon-CXP, Cyton-CXP Number of CXP down stream (camera to frame grabber) trigger rising edges.
<b>1_DT_TRIG_DELAY</b>	RO, CON123[23..16], Aon-CXP, Cyton-CXP The payload of the CXP down stream (camera to frame grabber) trigger.
<b>1_DT_FALL_DISABLE</b>	R/W, CON123[24], Aon-CXP, 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.
<b>1_DT_RISE_DISABLE</b>	R/W, CON123[25], Aon-CXP, 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.
<b>1_DT_CLR_CNT</b>	WO, CON123[30], Aon-CXP, Cyton-CXP Writing this register to 1 will clear that counters in this registers.
<b>1_DT_STATE</b>	RO, CON123[31], Aon-CXP, 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

<b>1_SERDES_STATE</b>	RO, CON159[7..0], Cyton-CXP, Karbon-CXP Current state of the SERDES associated with this link.
<b>1_LINK_SPEED</b>	RO, CON159[15..8], Cyton-CXP, Karbon-CXP Current speed of the SERDES associated with this link.
<b>1_LOST_ALIGN_CNT</b>	RO, CON159[25..16], Cyton-CXP, Karbon-CXP Number of times the SERDES associated with this link has lost alignment.
<b>1_RESYNC_TOG_MSTR</b>	RO, CON159[28], Cyton-CXP, Karbon-CXP Diagnostics.
<b>1_RESYNC_TOG</b>	RO, CON159[29], Cyton-CXP, Karbon-CXP Diagnostics.
<b>1_SERDES_ALIGNED</b>	RO, CON159[30], Cyton-CXP, Karbon-CXP If this bit is 1 the SERDES associated with this link is aligned.

## 10.38 CON160

<b>Bit</b>	<b>Name</b>
0	1_RAW_DATA_MODE
1	1_REMOVE_IDLEES
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\_RAW\_DATA\_MODE** R/W, CON160[0], Cyton-CXP  
See description of 0\_RAW\_DATA\_MODE

**1\_REMOVE\_IDLES** R/W, CON160[1], Cyton-CXP  
See description of 0\_REMOVE\_IDLES

## 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], Cyton-CXP

See description of 0\_SERDES\_ERROR\_CODE

## 10.40 CON168

Bit	Name
0	2_POEXP_EN_POWER
1	2_POEXP_EN_24V_REG
2	2_POEXP_EN_CAM_SENSE
3	2_POEXP_CAM_IS_POEXP
4	2_POEXP_SHORT_DETECTED
5	2_POEXP_OPEN_DETECTED
6	2_POEXP_OVER_DETECTED
7	2_POEXP_OVER_LATCH
8	2_POEXP_UNDER_DETECTED
9	2_POEXP_UNDER_LATCH
10	2_POEXP_24V_OK
11	Reserved
12	2_POEXP_STATE
13	2_POEXP_OVR_AUTO_RESTART
14	2_POEXP_SENSE_BYPASS
15	2_ENABLE_POEXP_SYSTEM
16	2_POEXP_CURRENT_LATCH
17	2_POEXP_CURRENT_LATCH
18	2_POEXP_CURRENT_LATCH
19	2_POEXP_CURRENT_LATCH
20	2_POEXP_CURRENT_LATCH
21	2_POEXP_CURRENT_LATCH
22	2_POEXP_CURRENT_LATCH
23	2_POEXP_CURRENT_LATCH
24	2_POEXP_CURRENT
25	2_POEXP_CURRENT
26	2_POEXP_CURRENT
27	2_POEXP_CURRENT
28	2_POEXP_CURRENT
29	2_POEXP_CURRENT
30	2_POEXP_CURRENT
31	2_POEXP_CURRENT



<b>2_POCPX_EN_POWER</b>	R/W, CON168[0], Cyton-CXP See description of 0_POCPX_EN_POWER
<b>2_POCPX_EN_24V_REG</b>	R/W, CON168[1], Cyton-CXP See description of 0_POCPX_EN_24V_REG
<b>2_POCPX_EN_CAM_SENSE</b>	RO, CON168[2], Cyton-CXP See description of 0_POCPX_EN_CAM_SENSE
<b>2_POCPX_CAM_IS_POCPX</b>	RO, CON168[3], Cyton-CXP See description of 0_POCPX_CAM_IS_POCPX
<b>2_POCPX_SHORT_DETECTED</b>	RO, CON168[4], Cyton-CXP See description of 0_POCPX_SHORT_DETECTED
<b>2_POCPX_OPEN_DETECTED</b>	RO, CON168[5], Cyton-CXP See description of 0_POCPX_OPEN_DETECTED
<b>2_POCPX_OVER_DETECTED</b>	RO, CON168[6], Cyton-CXP See description of 0_POCPX_OVER_DETECTED
<b>2_POCPX_OVER_LATCH</b>	RO, CON168[7], Cyton-CXP See description of 0_POCPX_OVER_LATCH
<b>2_POCPX_UNDER_DETECTED</b>	RO, CON168[8], Cyton-CXP See description of 0_POCPX_UNDER_DETECTED
<b>2_POCPX_UNDER_LATCH</b>	RO, CON168[9], Cyton-CXP See description of 0_POCPX_UNDER_LATCH

<b>2_POEXP_24V_OK</b>	RO, CON168[10], Cyton-CXP See description of 0_POEXP_24V_OK
<b>2_POEXP_STATE</b>	RO, CON168[12], Cyton-CXP See description of 0_POEXP_STATE.
<b>2_POEXP_OVR_AUTO_RESTART</b>	RW, CON168[13], Cyton-CXP See description of 0_POEXP_OVR_AUTO_RESTART.
<b>2_POEXP_SENSE_BYPASS</b>	RW, CON168[14], Cyton-CXP See description of 0_POEXP_SENSE_BYPASS.
<b>2_ENABLE_POEXP_SYSTEM</b>	RW, CON168[15], Cyton-CXP See description of 0_ENABLE_POEXP_SYSTEM.
<b>2_POEXP_CURRENT_LATCH</b>	RO, CON168[23..16], Cyton-CXP See description of 0_POEXP_CURRENT_LATCH
<b>2_POEXP_CURRENT</b>	RO, CON168[31..24], Cyton-CXP See description of 0_POEXP_CURRENT

## 10.41 CON169

<b>Bit</b>	<b>Name</b>
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\_POCPX\_  
OVER\_TIMER**

R/W, CON169[31..0], Cyton-CXP

See description of 0\_POCPX\_OVER\_TIMER

## 10.42 CON170

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

**2\_POCPX\_  
UNDER\_TIMER**

R/W, CON170[31..0], Cyton-CXP

See description of 0\_POCPX\_UNDER\_TIMER

## 10.43 CON171

<b>Bit</b>	<b>Name</b>
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], Cyton-CXP  
See description of 0\_COM\_RCV\_FIFO\_SIZE

**2\_COM\_SEND\_FIFO\_SIZE** RO, CON171[31..16], 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	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_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], Cyton-CXP  
See description of 0\_COM\_SEND\_FIFO\_CLR

**2\_COM\_SEND\_GO** WO, CON172[1], Cyton-CXP  
See description of 0\_COM\_SEND\_GO

**2\_COM\_SEND\_FIFO\_CNT** RO, CON172[31..16], Cyton-CXP  
See description of 0\_COM\_SEND\_FIFO\_CNT

## 10.45 CON173

<b>Bit</b>	<b>Name</b>
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], 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], Cyton-CXP

See description of 0\_COM\_RCV\_FIFO\_CLR

**2\_COM\_RCV\_  
FIFO\_CNT**

RO, CON174[31..16], Cyton-CXP

See description of 0\_COM\_RCV\_FIFO\_CNT

## 10.47 CON175

<b>Bit</b>	<b>Name</b>
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], Cyton-CXP

See description of 0\_COM\_RCV\_DATA



## 10.48 CON179

<b>Bit</b>	<b>Name</b>
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], 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_GPIO_ACK_RCVD
4	2_CTL_ACK_RCVD
5	2_GPIO_RCVD
6	2_TRIG_RCVD
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_IOACK_UNEXPECTED_INT
14	2_IOACK_NOMATCH2
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

<b>2_UNDER_CURRENT</b>	R/W, CON180[0], Cyton-CXP See description of 0_UNDER_CURRENT
<b>2_OVER_CURRENT</b>	R/W, CON180[1], Cyton-CXP See description of 0_OVER_CURRENT
<b>2_TRIG_ACK_RCVD</b>	R/W, CON180[2], Cyton-CXP See description of 0_TRIG_ACK_RCVD
<b>2_GPIO_ACK_RCVD</b>	R/W, CON180[3], Cyton-CXP See description of 0_GPIO_ACK_RCVD
<b>2_CTL_ACK_RCVD</b>	R/W, CON180[4], Cyton-CXP See description of 0_CTL_ACK_RCVD
<b>2_GPIO_RCVD</b>	R/W, CON180[5], Cyton-CXP See description of 0_GPIO_RCVD
<b>2_TRIG_RCVD</b>	R/W, CON180[6], Cyton-CXP See description of 0_TRIG_RCVD
<b>2_CTL_RSP_FIFO_OVF</b>	R/W, CON180[7], Cyton-CXP See description of 0_CTL_RSP_FIFO_OVF
<b>2_CTL_REQ_FIFO_OVF</b>	R/W, CON180[8], Cyton-CXP See description of 0_CTL_REQ_FIFO_OVF
<b>2_DOWN_TRIG_RCVD</b>	R/W, CON180[9], Cyton-CXP See description of 0_DOWN_TRIG_RCVD

<b>2_TRIG_NOMATCH</b>	R/W, CON180[10], Cyton-CXP See description of 0_TRIG_NOMATCH
<b>2_IOACK_UNKNOWN_TYPE</b>	R/W, CON180[11], Cyton-CXP See description of 0_IOACK_UNKNOWN_TYPE
<b>2_IOACK_NOMATCH</b>	R/W, CON180[12], Cyton-CXP See description of 0_IOACK_NOMATCH
<b>2_IOACK_UNEXPECTED_INT</b>	R/W, CON180[13], Cyton-CXP See description of 0_IOACK_UNEXPECTED_INT
<b>2_IOACK_NOMATCH2</b>	R/W, CON180[14], Cyton-CXP See description of 0_IOACK_NOMATCH2
<b>2_STRM_PKT_DROP</b>	R/W, CON180[15], Cyton-CXP See description of 0_STRM_PKT_DROP
<b>2_STRM_NOT_ENOUGH_DAT</b>	R/W, CON180[16], Cyton-CXP See description of 0_STRM_NOT_ENOUGH_DAT
<b>2_STRM_TOO_MUCH_DAT</b>	R/W, CON180[17], Cyton-CXP See description of 0_STRM_TOO_MUCH_DAT
<b>2_STRM_BAD_CRC</b>	R/W, CON180[18], Cyton-CXP See description of 0_STRM_BAD_CRC
<b>2_STRM_OVERFLOW</b>	R/W, CON180[19], Cyton-CXP See description of 0_STRM_OVERFLOW

## 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

<b>2_UNDER_CURRENT_M</b>	R/W, CON181[0], Cyton-CXP See description of 0_UNDER_CURRENT_M
<b>2_OVER_CURRENT_M</b>	R/W, CON181[1], Cyton-CXP See description of 0_OVER_CURRENT_M
<b>2_TRIG_ACK_RCVD_M</b>	R/W, CON181[2], Cyton-CXP See description of 0_TRIG_ACK_RCVD_M
<b>2_GPIO_ACK_RCVD_M</b>	R/W, CON181[3], Cyton-CXP See description of 0_GPIO_ACK_RCVD_M
<b>2_CTL_ACK_RCVD_M</b>	R/W, CON181[4], Cyton-CXP See description of 0_CTL_ACK_RCVD_M
<b>2_GPIO_RCVD_M</b>	R/W, CON181[5], Cyton-CXP See description of 0_GPIO_RCVD_M
<b>2_TRIG_RCVD_M</b>	R/W, CON181[6], Cyton-CXP See description of 0_TRIG_RCVD_M
<b>2_CTL_RSP_FIFO_OVF_M</b>	R/W, CON181[7], Cyton-CXP See description of 0_CTL_RSP_FIFO_OVF_M
<b>2_CTL_REQ_FIFO_OVF_M</b>	R/W, CON181[8], Cyton-CXP See description of 0_CTL_REQ_FIFO_OVF_M
<b>2_DOWN_TRIG_RCVD_M</b>	R/W, CON181[9], Cyton-CXP See description of 0_DOWN_TRIG_RCVD_M

<b>2_TRIG_NOMATCH_M</b>	R/W, CON181[10], Cyton-CXP See description of 0_TRIG_NOMATCH_M
<b>2_IOACK_UNKNOWN_TYPE_M</b>	R/W, CON181[11], Cyton-CXP See description of 0_IOACK_UNKNOWN_TYPE_M
<b>2_IOACK_NOMATCH_M</b>	R/W, CON181[12], Cyton-CXP See description of 0_IOACK_NOMATCH_M
<b>2_IOACK_UNEXPECTED_INT_M</b>	R/W, CON181[13], Cyton-CXP See description of 0_IOACK_UNEXPECTED_INT_M
<b>2_IOACK_NOMATCH2_M</b>	R/W, CON181[14], Cyton-CXP See description of 0_IOACK_NOMATCH2_M
<b>2_STRM_PKT_DROP_M</b>	R/W, CON181[15], Cyton-CXP See description of 0_STRM_PKT_DROP_M
<b>2_STRM_NOT_ENOUGH_DAT_M</b>	R/W, CON181[16], Cyton-CXP See description of 0_STRM_NOT_ENOUGH_DAT_M
<b>2_STRM_TOO_MUCH_DAT_M</b>	R/W, CON181[17], Cyton-CXP See description of 0_STRM_TOO_MUCH_DAT_M
<b>2_STRM_BAD_CRC_M</b>	R/W, CON181[18], Cyton-CXP See description of 0_STRM_BAD_CRC_M
<b>2_STRM_OVERFLOW_M</b>	R/W, CON181[19], 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], Cyton-CXP

See description of 0\_UNDER\_CURRENT\_WP

**2\_OVER\_CURRENT\_WP**

R/W, CON182[1], Cyton-CXP

See description of 0\_OVER\_CURRENT\_WP

**2\_TRIG\_ACK\_RCVD\_WP**

R/W, CON182[2], Cyton-CXP

See description of 0\_TRIG\_ACK\_RCVD\_WP

**2\_GPIO\_ACK\_RCVD\_WP**

R/W, CON182[3], Cyton-CXP

See description of 0\_GPIO\_ACK\_RCVD\_WP

**2\_CTL\_ACK\_RCVD\_WP**

R/W, CON182[4], Cyton-CXP

See description of 0\_CTL\_ACK\_RCVD\_WP

**2\_GPIO\_RCVD\_WP**

R/W, CON182[5], Cyton-CXP

See description of 0\_GPIO\_RCVD\_WP

**2\_TRIG\_RCVD\_WP**

R/W, CON182[6], Cyton-CXP

See description of 0\_TRIG\_RCVD\_WP

**2\_CTL\_RSP\_FIFO\_OVF\_WP**

R/W, CON182[7], Cyton-CXP

See description of 0\_CTL\_RSP\_FIFO\_OVF\_WP

**2\_CTL\_REQ\_FIFO\_OVF\_WP**

R/W, CON182[8], Cyton-CXP

See description of 0\_CTL\_REQ\_FIFO\_OVF\_WP

**2\_DOWN\_TRIG\_RCVD\_WP**

R/W, CON182[9], Cyton-CXP

See description of 0\_DOWN\_TRIG\_RCVD\_WP

<b>2_TRIG_NOMATCH_WP</b>	R/W, CON182[10], Cyton-CXP See description of 0_TRIG_NOMATCH_WP
<b>2_IOACK_UNKNOWN_TYPE_WP</b>	R/W, CON182[11], Cyton-CXP See description of 0_IOACK_UNKNOWN_TYPE_WP
<b>2_IOACK_NOMATCH_WP</b>	R/W, CON182[12], Cyton-CXP See description of 0_IOACK_NOMATCH_WP
<b>2_IOACK_UNEXPECTED_INT_WP</b>	R/W, CON182[13], Cyton-CXP See description of 0_IOACK_UNEXPECTED_INT_WP
<b>2_IOACK_NOMATCH2_WP</b>	R/W, CON182[14], Cyton-CXP See description of 0_IOACK_NOMATCH2_WP
<b>2_STRM_PKT_DROP_WP</b>	R/W, CON182[15], Cyton-CXP See description of 0_STRM_PKT_DROP_WP
<b>2_STRM_NOT_ENOUGH_DAT_WP</b>	R/W, CON182[16], Cyton-CXP See description of 0_STRM_NOT_ENOUGH_DAT_WP
<b>2_STRM_TOO_WPUCH_DAT_WP</b>	R/W, CON182[17], Cyton-CXP See description of 0_STRM_TOO_WPUCH_DAT_WP
<b>2_STRM_BAD_CRC_WP</b>	R/W, CON182[18], Cyton-CXP See description of 0_STRM_BAD_CRC_WP
<b>2_STRM_OVERFLOW_WP</b>	R/W, CON182[19], 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], Cyton-CXP  
See description of 0\_PKT\_RCVD\_CNT

**2\_PKT\_GNT\_CNT** RO, CON184[31..16], 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], Cyton-CXP  
See description of 0\_PKT\_DROP\_CNT

**2\_CRC\_ERR\_CNT** RO, CON185[31..16], 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



<b>2_CXP_TRIG_FALL_SENT</b>	RO, CON186[7..0], Cyton-CXP Number of CXP trigger falling edge trigger packets sent to the camera.
<b>2_CXP_TRIG_RISE_SENT</b>	RO, CON186[15..8], Cyton-CXP Number of CXP trigger rising edge trigger packets sent to the camera.
<b>2_CXP_TRIG_ACK_CNT</b>	RO, CON186[23..16], Cyton-CXP Number of CXP trigger acknowledgments received from the camera.
<b>2_CXP_TRIG_STATE</b>	RO, CON186[31], 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	Reserved
27	Reserved
28	Reserved
29	Reserved
30	2_DT_CLR_CNT
31	2_DT_STATE

<b>2_DT_FALL_RCVD_COUNT</b>	RO, CON123[7..0], Aon-CXP, Cyton-CXP Number of CXP down stream (camera to frame grabber) trigger falling edges.
<b>2_DT_RISE_RCVD_COUNT</b>	RO, CON123[15..8], Aon-CXP, Cyton-CXP Number of CXP down stream (camera to frame grabber) trigger rising edges.
<b>2_DT_TRIG_DELAY</b>	RO, CON123[23..16], Aon-CXP, Cyton-CXP The payload of the CXP down stream (camera to frame grabber) trigger.
<b>2_DT_FALL_DISABLE</b>	R/W, CON123[24], Aon-CXP, 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.
<b>2_DT_RISE_DISABLE</b>	R/W, CON123[25], Aon-CXP, 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.
<b>2_DT_CLR_CNT</b>	WO, CON123[30], Aon-CXP, Cyton-CXP Writing this register to 1 will clear that counters in this registers.
<b>2_DT_STATE</b>	RO, CON123[31], Aon-CXP, 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

<b>2_SERDES_STATE</b>	RO, CON191[7..0], Cyton-CXP, Karbon-CXP Current state of the SERDES associated with this link.
<b>2_LINK_SPEED</b>	RO, CON191[15..8], Cyton-CXP, Karbon-CXP Current speed of the SERDES associated with this link.
<b>2_LOST_ALIGN_CNT</b>	RO, CON191[25..16], Cyton-CXP, Karbon-CXP Describe 2_LOST_ALIGN_CNT here.
<b>2_RESYNC_TOG_MSTR</b>	RO, CON191[28], Cyton-CXP, Karbon-CXP Diagnostics.
<b>2_RESYNC_TOG</b>	RO, CON191[29], Cyton-CXP, Karbon-CXP Diagnostics.
<b>2_SERDES_ALIGNED</b>	RO, CON191[30], Cyton-CXP, Karbon-CXP If this bit is 1 the SERDES associated with this link is aligned.

## 10.57 CON192

<b>Bit</b>	<b>Name</b>
0	2_RAW_DATA_MODE
1	2_REMOVE_IDLEES
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\_RAW\_DATA\_MODE** R/W, CON192[0], Cyton-CXP  
See description of 0\_RAW\_DATA\_MODE

**2\_REMOVE\_IDLES** R/W, CON192[1], Cyton-CXP  
See description of 0\_REMOVE\_IDLES

## 10.58 CON195

<b>Bit</b>	<b>Name</b>
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], Cyton-CXP

See description of 0\_SERDES\_ERROR\_CODE

## 10.59 CON200

Bit	Name
0	3_POEXP_EN_POWER
1	3_POEXP_EN_24V_REG
2	3_POEXP_EN_CAM_SENSE
3	3_POEXP_CAM_IS_POEXP
4	3_POEXP_SHORT_DETECTED
5	3_POEXP_OPEN_DETECTED
6	3_POEXP_OVER_DETECTED
7	3_POEXP_OVER_LATCH
8	3_POEXP_UNDER_DETECTED
9	3_POEXP_UNDER_LATCH
10	3_POEXP_24V_OK
11	Reserved
12	3_POEXP_STATE
13	3_POEXP_OVR_AUTO_RESTART
14	3_POEXP_SENSE_BYPASS
15	3_ENABLE_POEXP_SYSTEM
16	3_POEXP_CURRENT_LATCH
17	3_POEXP_CURRENT_LATCH
18	3_POEXP_CURRENT_LATCH
19	3_POEXP_CURRENT_LATCH
20	3_POEXP_CURRENT_LATCH
21	3_POEXP_CURRENT_LATCH
22	3_POEXP_CURRENT_LATCH
23	3_POEXP_CURRENT_LATCH
24	3_POEXP_CURRENT
25	3_POEXP_CURRENT
26	3_POEXP_CURRENT
27	3_POEXP_CURRENT
28	3_POEXP_CURRENT
29	3_POEXP_CURRENT
30	3_POEXP_CURRENT
31	3_POEXP_CURRENT

<b>3_POCPX_EN_POWER</b>	RO, CON200[0], Cyton-CXP See description of 0_POCPX_EN_POWER
<b>3_POCPX_EN_24V_REG</b>	RO, CON200[1], Cyton-CXP See description of 0_POCPX_EN_24V_REG
<b>3_POCPX_EN_CAM_SENSE</b>	RO, CON200[2], Cyton-CXP See description of 0_POCPX_EN_CAM_SENSE
<b>3_POCPX_CAM_IS_POCPX</b>	RO, CON200[3], Cyton-CXP See description of 0_POCPX_CAM_IS_POCPX
<b>3_POCPX_SHORT_DETECTED</b>	RO, CON200[4], Cyton-CXP See description of 0_POCPX_SHORT_DETECTED
<b>3_POCPX_OPEN_DETECTED</b>	RO, CON200[5], Cyton-CXP See description of 0_POCPX_OPEN_DETECTED
<b>3_POCPX_OVER_DETECTED</b>	RO, CON200[6], Cyton-CXP See description of 0_POCPX_OVER_DETECTED
<b>3_POCPX_OVER_LATCH</b>	RO, CON200[7], Cyton-CXP See description of 0_POCPX_OVER_LATCH
<b>3_POCPX_UNDER_DETECTED</b>	RO, CON200[8], Cyton-CXP See description of 0_POCPX_UNDER_DETECTED
<b>3_POCPX_UNDER_LATCH</b>	RO, CON200[9], Cyton-CXP See description of 0_POCPX_UNDER_LATCH

<b>3_POEXP_24V_OK</b>	RO, CON200[10], Cyton-CXP See description of 0_POEXP_24V_OK
<b>3_POEXP_STATE</b>	RO, CON200[12], Cyton-CXP See description of 0_POEXP_STATE.
<b>3_POEXP_OVR_AUTO_RESTART</b>	RW, CON200[13], Cyton-CXP See description of 0_POEXP_OVR_AUTO_RESTART.
<b>3_POEXP_SENSE_BYPASS</b>	RW, CON200[14], Cyton-CXP See description of 0_POEXP_SENSE_BYPASS.
<b>3_ENABLE_POEXP_SYSTEM</b>	RW, CON200[15], Cyton-CXP See description of 0_ENABLE_POEXP_SYSTEM.
<b>3_POEXP_CURRENT_LATCH</b>	RO, CON200[23..16], Cyton-CXP See description of 0_POEXP_CURRENT_LATCH
<b>3_POEXP_CURRENT</b>	RO, CON200[31..24], Cyton-CXP See description of 0_POEXP_CURRENT

## 10.60 CON201

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

**3\_POEXP\_  
OVER\_TIMER**

R/W, CON201[31..0], Cyton-CXP

See description of 0\_POEXP\_OVER\_TIMER

## 10.61 CON202

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

**3\_POCPX\_  
UNDER\_TIMER**

R/W, CON202[31..0], Cyton-CXP

See description of 0\_POCPX\_UNDER\_TIMER



## 10.62 CON203

<b>Bit</b>	<b>Name</b>
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], Cyton-CXP  
See description of 0\_COM\_RCV\_FIFO\_SIZE

**3\_COM\_SEND\_FIFO\_SIZE** RO, CON203[31..16], 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	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_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], Cyton-CXP  
See description of 0\_COM\_SEND\_FIFO\_CLR

**3\_COM\_SEND\_GO** WO, CON204[1], Cyton-CXP  
See description of 0\_COM\_SEND\_GO

**3\_COM\_SEND\_FIFO\_CNT** RO, CON204[31..16], Cyton-CXP  
See description of 0\_COM\_SEND\_FIFO\_CNT

## 10.64 CON205

<b>Bit</b>	<b>Name</b>
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], Cyton-CXP  
See description of 0\_COM\_SEND\_DATA

## 10.65 CON206

<b>Bit</b>	<b>Name</b>
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], Cyton-CXP

See description of 0\_COM\_RCV\_FIFO\_CLR

**3\_COM\_RCV\_  
FIFO\_CNT**

RO, CON206[31..16], Cyton-CXP

See description of 0\_COM\_RCV\_FIFO\_CNT



## 10.66 CON207

<b>Bit</b>	<b>Name</b>
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], Cyton-CXP

See description of 0\_COM\_RCV\_DATA

## 10.67 CON211

<b>Bit</b>	<b>Name</b>
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], 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_GPIO_ACK_RCVD
4	3_CTL_ACK_RCVD
5	3_GPIO_RCVD
6	3_TRIG_RCVD
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_IOACK_UNEXPECTED_INT
14	3_IOACK_NOMATCH2
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

<b>3_UNDER_CURRENT</b>	R/W, CON212[0], Cyton-CXP See description of 0_UNDER_CURRENT
<b>3_OVER_CURRENT</b>	R/W, CON212[1], Cyton-CXP See description of 0_OVER_CURRENT
<b>3_TRIG_ACK_RCVD</b>	R/W, CON212[2], Cyton-CXP See description of 0_TRIG_ACK_RCVD
<b>3_GPIO_ACK_RCVD</b>	R/W, CON212[3], Cyton-CXP See description of 0_GPIO_ACK_RCVD
<b>3_CTL_ACK_RCVD</b>	R/W, CON212[4], Cyton-CXP See description of 0_CTL_ACK_RCVD
<b>3_GPIO_RCVD</b>	R/W, CON212[5], Cyton-CXP See description of 0_GPIO_RCVD
<b>3_TRIG_RCVD</b>	R/W, CON212[6], Cyton-CXP See description of 0_TRIG_RCVD
<b>3_CTL_RSP_FIFO_OVF</b>	R/W, CON212[7], Cyton-CXP See description of 0_CTL_RSP_FIFO_OVF
<b>3_CTL_REQ_FIFO_OVF</b>	R/W, CON212[8], Cyton-CXP See description of 0_CTL_REQ_FIFO_OVF
<b>3_DOWN_TRIG_RCVD</b>	R/W, CON212[9], Cyton-CXP See description of 0_DOWN_TRIG_RCVD

<b>3_TRIG_NOMATCH</b>	R/W, CON212[10], Cyton-CXP See description of 0_TRIG_NOMATCH
<b>3_IOACK_UNKNOWN_TYPE</b>	R/W, CON212[11], Cyton-CXP See description of 0_IOACK_UNKNOWN_TYPE
<b>3_IOACK_NOMATCH</b>	R/W, CON212[12], Cyton-CXP See description of 0_IOACK_NOMATCH
<b>3_IOACK_UNEXPECTED_INT</b>	R/W, CON212[13], Cyton-CXP See description of 0_IOACK_UNEXPECTED_INT
<b>3_IOACK_NOMATCH2</b>	R/W, CON212[14], Cyton-CXP See description of 0_IOACK_NOMATCH2
<b>3_STRM_PKT_DROP</b>	R/W, CON212[15], Cyton-CXP See description of 0_STRM_PKT_DROP
<b>3_STRM_NOT_ENOUGH_DAT</b>	R/W, CON212[16], Cyton-CXP See description of 0_STRM_NOT_ENOUGH_DAT
<b>3_STRM_TOO_MUCH_DAT</b>	R/W, CON212[17], Cyton-CXP See description of 0_STRM_TOO_MUCH_DAT
<b>3_STRM_BAD_CRC</b>	R/W, CON212[18], Cyton-CXP See description of 0_STRM_BAD_CRC
<b>3_STRM_OVERFLOW</b>	R/W, CON212[19], Cyton-CXP See description of 0_STRM_OVERFLOW

## 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



<b>3_UNDER_CURRENT_M</b>	R/W, CON213[0], Cyton-CXP See description of 0_UNDER_CURRENT_M
<b>3_OVER_CURRENT_M</b>	R/W, CON213[1], Cyton-CXP See description of 0_OVER_CURRENT_M
<b>3_TRIG_ACK_RCVD_M</b>	R/W, CON213[2], Cyton-CXP See description of 0_TRIG_ACK_RCVD_M
<b>3_GPIO_ACK_RCVD_M</b>	R/W, CON213[3], Cyton-CXP See description of 0_GPIO_ACK_RCVD_M
<b>3_CTL_ACK_RCVD_M</b>	R/W, CON213[4], Cyton-CXP See description of 0_CTL_ACK_RCVD_M
<b>3_GPIO_RCVD_M</b>	R/W, CON213[5], Cyton-CXP See description of 0_GPIO_RCVD_M
<b>3_TRIG_RCVD_M</b>	R/W, CON213[6], Cyton-CXP See description of 0_TRIG_RCVD_M
<b>3_CTL_RSP_FIFO_OVF_M</b>	R/W, CON213[7], Cyton-CXP See description of 0_CTL_RSP_FIFO_OVF_M
<b>3_CTL_REQ_FIFO_OVF_M</b>	R/W, CON213[8], Cyton-CXP See description of 0_CTL_REQ_FIFO_OVF_M
<b>3_DOWN_TRIG_RCVD_M</b>	R/W, CON213[9], Cyton-CXP See description of 0_DOWN_TRIG_RCVD_M

<b>3_TRIG_NOMATCH_M</b>	R/W, CON213[10], Cyton-CXP See description of 0_TRIG_NOMATCH_M
<b>3_IOACK_UNKNOWN_TYPE_M</b>	R/W, CON213[11], Cyton-CXP See description of 0_IOACK_UNKNOWN_TYPE_M
<b>3_IOACK_NOMATCH_M</b>	R/W, CON213[12], Cyton-CXP See description of 0_IOACK_NOMATCH_M
<b>3_IOACK_UNEXPECTED_INT_M</b>	R/W, CON213[13], Cyton-CXP See description of 0_IOACK_UNEXPECTED_INT_M
<b>3_IOACK_NOMATCH2_M</b>	R/W, CON213[14], Cyton-CXP See description of 0_IOACK_NOMATCH2_M
<b>3_STRM_PKT_DROP_M</b>	R/W, CON213[15], Cyton-CXP See description of 0_STRM_PKT_DROP_M
<b>3_STRM_NOT_ENOUGH_DAT_M</b>	R/W, CON213[16], Cyton-CXP See description of 0_STRM_NOT_ENOUGH_DAT_M
<b>3_STRM_TOO_MUCH_DAT_M</b>	R/W, CON213[17], Cyton-CXP See description of 0_STRM_TOO_MUCH_DAT_M
<b>3_STRM_BAD_CRC_M</b>	R/W, CON213[18], Cyton-CXP See description of 0_STRM_BAD_CRC_M
<b>3_STRM_OVERFLOW_M</b>	R/W, CON213[19], 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], Cyton-CXP

See description of 0\_UNDER\_CURRENT\_WP

**3\_OVER\_CURRENT\_WP**

R/W, CON214[1], Cyton-CXP

See description of 0\_OVER\_CURRENT\_WP

**3\_TRIG\_ACK\_RCVD\_WP**

R/W, CON214[2], Cyton-CXP

See description of 0\_TRIG\_ACK\_RCVD\_WP

**3\_GPIO\_ACK\_RCVD\_WP**

R/W, CON214[3], Cyton-CXP

See description of 0\_GPIO\_ACK\_RCVD\_WP

**3\_CTL\_ACK\_RCVD\_WP**

R/W, CON214[4], Cyton-CXP

See description of 0\_CTL\_ACK\_RCVD\_WP

**3\_GPIO\_RCVD\_WP**

R/W, CON214[5], Cyton-CXP

See description of 0\_GPIO\_RCVD\_WP

**3\_TRIG\_RCVD\_WP**

R/W, CON214[6], Cyton-CXP

See description of 0\_TRIG\_RCVD\_WP

**3\_CTL\_RSP\_FIFO\_OVF\_WP**

R/W, CON214[7], Cyton-CXP

See description of 0\_CTL\_RSP\_FIFO\_OVF\_WP

**3\_CTL\_REQ\_FIFO\_OVF\_WP**

R/W, CON214[8], Cyton-CXP

See description of 0\_CTL\_REQ\_FIFO\_OVF\_WP

**3\_DOWN\_TRIG\_RCVD\_WP**

R/W, CON214[9], Cyton-CXP

See description of 0\_DOWN\_TRIG\_RCVD\_WP

<b>3_TRIG_NOMATCH_WP</b>	R/W, CON214[10], Cyton-CXP See description of 0_TRIG_NOMATCH_WP
<b>3_IOACK_UNKNOWN_TYPE_WP</b>	R/W, CON214[11], Cyton-CXP See description of 0_IOACK_UNKNOWN_TYPE_WP
<b>3_IOACK_NOMATCH_WP</b>	R/W, CON214[12], Cyton-CXP See description of 0_IOACK_NOMATCH_WP
<b>3_IOACK_UNEXPECTED_INT_WP</b>	R/W, CON214[13], Cyton-CXP See description of 0_IOACK_UNEXPECTED_INT_WP
<b>3_IOACK_NOMATCH2_WP</b>	R/W, CON214[14], Cyton-CXP See description of 0_IOACK_NOMATCH2_WP
<b>3_STRM_PKT_DROP_WP</b>	R/W, CON214[15], Cyton-CXP See description of 0_STRM_PKT_DROP_WP
<b>3_STRM_NOT_ENOUGH_DAT_WP</b>	R/W, CON214[16], Cyton-CXP See description of 0_STRM_NOT_ENOUGH_DAT_WP
<b>3_STRM_TOO_WPUCH_DAT_WP</b>	R/W, CON214[17], Cyton-CXP See description of 0_STRM_TOO_WPUCH_DAT_WP
<b>3_STRM_BAD_CRC_WP</b>	R/W, CON214[18], Cyton-CXP See description of 0_STRM_BAD_CRC_WP
<b>3_STRM_OVERFLOW_WP</b>	R/W, CON214[19], Cyton-CXP See description of 0_STRM_OVERFLOW_WP

## 10.71 CON216

<b>Bit</b>	<b>Name</b>
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

<b>3_PKT_RCVD_CNT</b>	RO, CON216[15..0], Cyton-CXP See description of 0_PKT_RCVD_CNT
<b>3_PKT_GNT_CNT</b>	RO, CON216[31..16], 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], Cyton-CXP  
See description of 0\_PKT\_DROP\_CNT

**3\_CRC\_ERR\_CNT** RO, CON217[31..16], 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

<b>3_CXP_TRIG_FALL_SENT</b>	RO, CON218[7..0], Cyton-CXP Number of CXP trigger falling edge trigger packets sent to the camera.
<b>3_CXP_TRIG_RISE_SENT</b>	RO, CON218[15..8], Cyton-CXP Number of CXP trigger rising edge trigger packets sent to the camera.
<b>3_CXP_TRIG_ACK_CNT</b>	RO, CON218[23..16], Cyton-CXP Number of CXP trigger acknowledgments received from the camera.
<b>3_CXP_TRIG_STATE</b>	RO, CON218[31], 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], Cyton-CXP

See description of 0\_CXP\_TRIG\_STATE

**3\_CXP\_TRIG\_ACK\_CNT**

RO, CON218[31..24], 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	Reserved
27	Reserved
28	Reserved
29	Reserved
30	3_DT_CLR_CNT
31	3_DT_STATE

<b>3_DT_FALL_RCVD_COUNT</b>	RO, CON123[7..0], Aon-CXP, Cyton-CXP Number of CXP down stream (camera to frame grabber) trigger falling edges.
<b>3_DT_RISE_RCVD_COUNT</b>	RO, CON123[15..8], Aon-CXP, Cyton-CXP Number of CXP down stream (camera to frame grabber) trigger rising edges.
<b>3_DT_TRIG_DELAY</b>	RO, CON123[23..16], Aon-CXP, Cyton-CXP The payload of the CXP down stream (camera to frame grabber) trigger.
<b>3_DT_FALL_DISABLE</b>	R/W, CON123[24], Aon-CXP, 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.
<b>3_DT_RISE_DISABLE</b>	R/W, CON123[25], Aon-CXP, 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.
<b>3_DT_CLR_CNT</b>	WO, CON123[30], Aon-CXP, Cyton-CXP Writing this register to 1 will clear that counters in this registers.
<b>3_DT_STATE</b>	RO, CON123[31], Aon-CXP, 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



<b>3_SERDES_STATE</b>	RO, CON223[7..0], Cyton-CXP, Karbon-CXP Current state of the SERDES associated with this link.
<b>3_LINK_SPEED</b>	RO, CON223[15..8], Cyton-CXP, Karbon-CXP Current speed of the SERDES associated with this link.
<b>3_LOST_ALIGN_CNT</b>	RO, CON223[25..16], Cyton-CXP, Karbon-CXP Number of times the SERDES associated with this link has lost alignment.
<b>3_RESYNC_TOG_MSTR</b>	RO, CON223[28], Cyton-CXP, Karbon-CXP Diagnostics.
<b>3_RESYNC_TOG</b>	RO, CON223[29], Cyton-CXP, Karbon-CXP Diagnostics.
<b>3_SERDES_ALIGNED</b>	RO, CON223[30], Cyton-CXP, Karbon-CXP If this bit is 1 the SERDES associated with this link is aligned.

## 10.77 CON224

<b>Bit</b>	<b>Name</b>
0	3_RAW_DATA_MODE
1	3_REMOVE_IDLEES
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\_RAW\_DATA\_MODE** R/W, CON224[0], Cyton-CXP  
See description of 0\_RAW\_DATA\_MODE

**3\_REMOVE\_IDLES** R/W, CON224[1], Cyton-CXP  
See description of 0\_REMOVE\_IDLES

## 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], 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

<b>FW_BUILD_YEAR</b>	RO, CON356[15..0], Cyton-CXP Year that this firmware was compiled in BCD format. Example: 0x2012 is year 2012
<b>FW_BUILD_DAY</b>	RO, CON356[23..16], Cyton-CXP Day that this firmware was compiled in BCD format. Example: 0x18 the 18th of the month.
<b>FW_BUILD_MONTH</b>	RO, CON356[31..24], Cyton-CXP Month that this firmware was compiled in BCD format. Example: 0x12 is december.

## 10.80 CON357

<b>Bit</b>	<b>Name</b>
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



<b>FW_BUILD_MIN</b>	RO, CON357[7..0], Cyton-CXP
	Minute that this firmware was compiled. Example: 0x35 is 35 minutes past the hour.
<b>FW_BUILD_HOUR</b>	RO, CON357[15..8], Cyton-CXP
	Hour that this firmware was compiled. Example: 0x23 is 11pm (23rd hour).
<b>FPGA_ID</b>	RO, CON357[23..16], Cyton-CXP
	FPGA Identifier
<b>FW_CMPTBL</b>	RO, CON357[31..24], Cyton-CXP
	Firmware compatibility version (must match SDK driver internal firmware version).

## 10.81 CON358

<b>Bit</b>	<b>Name</b>
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	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

**PFG\_RESET**

R/W, CON358[0], Cyton-CXP

Reset the pfg (Stream assembler). This register does not get cleared automatically. Software must clear it.

## 10.82 CON360

<b>Bit</b>	<b>Name</b>
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], 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.

## 10.83 CON361

<b>Bit</b>	<b>Name</b>
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], 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], 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], 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], 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.

## 10.84 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)

## 10.85 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

<b>PKT_GNT_CNT</b>	RO, CON372[15..0], Cyton-CXP
	Total number of packets accepted from all links associated with this stream decoder. The counter wraps and is for debug only.
<b>NEXT_LINK</b>	RO, CON372[18..16], Cyton-CXP
	The next link from which a packet is expected.
<b>DISABLE_PKT_TAG_CHK</b>	RO, CON372[19], Cyton-CXP
	Disable the checking of packet tags.
<b>UNEXP_PKT_TAG_CNT</b>	RO, CON372[27..20], Cyton-CXP
	Number of unexpected packet tags received. These are out-of-order tags. The counter wraps and is for debug only.
<b>ACTIVE</b>	RO, CON372[31], Cyton-CXP
	The stream decoder is currently accepting a packet.

## 10.86 CON428

<b>Bit</b>	<b>Name</b>
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

<b>FORCE_RESYNC</b>	WO, CON428[0], Cyton-CXP Forces re-sync of CXP engine.
<b>PACK_PIXELS</b>	WO, CON428[1], Cyton-CXP Denotes that pixels will be DMAed packed.
<b>STRM_DECODE_STATE</b>	RO, CON428[6..4], Cyton-CXP Current state CXP stream decode engine.
<b>SOF_ERR</b>	RO, CON428[8], Cyton-CXP Bad Start Of Frame Packet.
<b>SOL_ERR</b>	RO, CON428[9], Cyton-CXP Bad Start Of Line Packet
<b>IH_ERR</b>	RO, CON428[10], Cyton-CXP Bad Image Header.
<b>SRC_TAG_ERR</b>	RO, CON428[11], Cyton-CXP Unexpected Source Tag error.
<b>IS_LINE_SCAN</b>	RO, CON428[12], Cyton-CXP Camera is a line scan camera.



# System Probe

---

## Chapter 11

### 11.1 Introduction

The System Probe is a circuit on the Aon/Axion/Cyton that can be used to measure the time between two hardware based events. A wide range of hardware events are available for measurements, see Table 11-1 for complete list. In addition to measure the time between two events, the System probe can also be set up to record some statistics. For example, the minimum time between two events can be recorded. These statistics can be taken over a period of time that begins on one event and ends on another event. Through some simple programming, a wide variety of complex phenomena can be measured.

The System Probe circuit is independent of any other section of the board, as such, it will not interfere with acquisition and DMA, nor do these blocks need to be programmed in order to measure events (except when measure events that are produced by these sections).

Two items are required to define an event: The source of the event (see table Table 11-1) and the event "function" (e.g. rising edge, falling edge, etc). To start measurement, SP\_ARM is written to 1. Another event can be use to "sync" measurement, this even it programmed into SP\_SYNC\_EVENT. Next, an event is used to start measurement (SP\_START\_EVENT). If not syn event is needed SP\_SYNC\_EVENT is programmed to be the same as SP\_START\_EVENT. Measure will proceed until SP\_STOP\_EVENT\_LIMIT number of events of type SP\_STOP\_EVENT have occurred. When the system is measuring events, SP\_BUSY reads back 1. When measurement finished, SP\_BUSY reads back 0, and the event total can be read from SP\_COUNT.

## 11.2 Example - Measure CL Clock Frequency

In this example we will set the System Probe up to measure the Camera Link clock frequency (Axion only). In this case, there will be no sync or start events, the recording will start immediately. The stop event will be the rising edge of a 100 nanosecond pulse train. The measurement will run until 1000000 of these 100 nanosecond pulses have occurred.

The System Probe registers must be programmed as shown in the following table.

Register	Value
SP_STOP_EVENT_LIMIT	1000000
SP_SYNC_EVENT	SPE_IMMEDIATE
SP_SYNC_FUNC	SPF_RISE
SP_START_EVENT	SPE_IMMEDIATE
SP_START_FUNC	SPF_RISE
SP_STOP_EVENT	SPE_100_NS
SP_STOP_FUNC	SPF_RISE
SP_TARGET_EVENT	SPE_CL_CLK_B
SP_TARGET_FUNC	SPF_RISE
SP_COUNT_MODE	1

Measurement will start when SP\_ARM is set to 1. While the measurement is taking place, SP\_BUSY will read back 1. When the measurement finished, SP\_BUSY will read back 0, and the number of events can be read out of SP\_COUNT.



### 11.3 Example - Measure LVALs Per Frame

In this example we will set the System Probe up to measure the number of lines per frame (LVALs per FVAL, Axion only). In this case, there will be no sync event. The start event will be the rising edge of FVAL and the stop event will be the falling edge of FVAL. The event measured will be the rising edge of LVAL. The measurement will run for 1 frame.

The System Probe registers must be programmed as shown in the following table.

Register	Value
SP_STOP_EVENT_LIMIT	1
SP_SYNC_EVENT	SPE_IMMEDIATE
SP_SYNC_FUNC	SPF_RISE
SP_START_EVENT	SPE_FVAL_B
SP_START_FUNC	SPF_RISE
SP_STOP_EVENT	SPE_FVAL_B
SP_STOP_FUNC	SPF_FALL
SP_TARGET_EVENT	SPE_CL_CLK_B
SP_TARGET_FUNC	SPE_LVAL_B
SP_COUNT_MODE	1

Measurement will start when SP\_ARM is set to 1. While the measurement is taking place, SP\_BUSY will read back 1. When the measurement finished, SP\_BUSY will read back 0, and the number of events can be read out of SP\_COUNT.

## 11.4 List of Events

Table 11-1 below enumerates all of the events that can be used for the System Probe. These events can be use for SP\_START\_EVENT, SP\_STOP\_EVENT, SP\_TARGET\_EVENT and SP\_SYNC\_EVENT.

*Note: Not all events are available on all models of frame grabber.*

**Table 11-1 System Probe Events**

<b>Event Name</b>	<b>Value</b>	<b>Comment</b>
SPE_NEVER	0x00	No Event - Never fires
SPE_IMMEDIATE	0x01	Immediate - Fires every SYS_CLK. same as SYS_CLK. Redundant but included anyways because the label IMMEDIATE is more intuitive
SPE_SYS_CLK	0x02	125MHz system clock
SPE_100_NS	0x03	100 nanosecond pulse
SPE_TSAUX_0	0x04	Timing Sequencer Aux 0
SPE_TSAUX_1	0x05	Timing Sequencer Aux 0
SPE_TSAUX_2	0x06	Timing Sequencer Aux 0
SPE_TSAUX_3	0x07	Timing Sequencer Aux 0
SPE_TRIG	0x08	System trigger
SPE_ENCA	0x09	Encoder A
SPE_ENCB	0x0a	Encoder B
SPE_ENCQ	0x0b	Encoder Q
SPE_ENCDIV	0x0c	Encoder Div
SPE_PCI_SENT	0x10	PCI packets sent (16-byte packets)
SPE_PCI_DROP	0x11	PCI packets dropped
SPE_PCI_BUSY	0x12	PCIe bus towards the host is busy
SPE_INTERRUPT	0x0f	System interrupt for this VFG
SPE_CL_CLK_B	0x20	C L Clock Base Channel
SPE_CL_CLK_M	0x21	CL Clock Med Channel
SPE_CL_CLK_F	0x22	CL Clock Full Channel
SPE_PLL_LOCK_B	0x23	PLL Lock Base Channel
SPE_PLL_LOCK_M	0x24	PLL Lock Med Channel
SPE_PLL_LOCK_F	0x25	PLL Lock Full Channel

Table 11-1 System Probe Events

<b>Event Name</b>	<b>Value</b>	<b>Comment</b>
SPE_FVAL_B	0x26	FVAL Base Channel
SPE_FVAL_M	0x27	FVAL Med Channel
SPE_FVAL_F	0x28	FVAL Full Channel
SPE_LVAL_B	0x29	LVAL Base Channel
SPE_LVAL_M	0x2A	LVAL Med Channel
SPE_LVAL_F	0x2B	LVAL Full Channel
SPE_DVAL_B	0x2C	DVAL Base Channel
SPE_DVAL_M	0x2D	DVAL Med Channel
SPE_DVAL_F	0x2E	DVAL Full Channel
SPE_CC1	0x2F	CC1
SPE_CC2	0x30	CC2
SPE_CC3	0x31	CC3
SPE_CC4	0x32	CC3

## 11.5 SP\_EVENTS

Bit	Name
0	SP_TARGET_EVENT
1	SP_TARGET_EVENT
2	SP_TARGET_EVENT
3	SP_TARGET_EVENT
4	SP_TARGET_EVENT
5	SP_TARGET_EVENT
6	SP_TARGET_FUNC
7	SP_TARGET_FUNC
8	SP_STOP_EVENT
9	SP_STOP_EVENT
10	SP_STOP_EVENT
11	SP_STOP_EVENT
12	SP_STOP_EVENT
13	SP_STOP_EVENT
14	SP_STOP_FUNC
15	SP_STOP_FUNC
16	SP_START_EVENT
17	SP_START_EVENT
18	SP_START_EVENT
19	SP_START_EVENT
20	SP_START_EVENT
21	SP_START_EVENT
22	SP_START_FUNC
23	SP_START_FUNC
24	SP_SYNC_EVENT
25	SP_SYNC_EVENT
26	SP_SYNC_EVENT
27	SP_SYNC_EVENT
28	SP_SYNC_EVENT
29	SP_SYNC_EVENT
30	SP_SYNC_FUNC
31	SP_SYNC_FUNC

**SP\_TARGET\_EVENT**

R/W, SP\_EVENTS[5..0], Aon-CXP, Axion-CL, Cyton-CXP

Target events to count between start and stop. See Table 11-1 for a list of events and values

**SP\_TARGET\_FUNC**

R/W, SP\_EVENTS[7..6], Aon-CXP, Axion-CL, Cyton-CXP

Function of the event to count.

SP_TARGET_FUNC	Meaning
0 (00b)	Rising edge
1 (01b)	Falling edge
2 (10b)	High
3 (11b)	Low

**SP\_STOP\_EVENT** R/W, SP\_EVENTS[13..8], Aon-CXP, Axion-CL, Cyton-CXP

Event that ends the counting. SP\_COUNT will be updated with the target count (or some function of it based on SP\_COUNT\_MODE). See Table 11-1 for a list of events and values

**SP\_STOP\_FUNC** R/W, SP\_EVENTS[15..14], Aon-CXP, Axion-CL, Cyton-CXP

Function of the stop event use to close the event window.

SP_STOP_FUNC	Meaning
0 (00b)	Rising edge
1 (01b)	Falling edge
2 (10b)	High
3 (11b)	Low

**SP\_START\_EVENT**

R/W, SP\_EVENTS[21..16], Aon-CXP, Axion-CL, Cyton-CXP

Event that starts the counting.

**SP\_START\_FUNC** R/W, SP\_EVENTS[23..22], Aon-CXP, Axion-CL, Cyton-CXP

Function of the start event that starts the counting. See Table 11-1 for a list of events and values

SP_START_FUNC	Meaning
0 (00b)	Rising edge
1 (01b)	Falling edge
2 (10b)	High
3 (11b)	Low

**SP\_SYNC\_EVENT** R/W, SP\_EVENTS[29..24], Aon-CXP, Axion-CL, Cyton-CXP

In situations where another event is needed to sync the counting this event can be used. If not sync event is needed SP\_SYNC\_EVENT and SP\_START\_EVENT can be the same. See Table 11-1 for a list of events and values

**SP\_SYNC\_FUNC** R/W, SP\_EVENTS[31..30], Aon-CXP, Axion-CL, Cyton-CXP

Function of the sync event that starts the counting.

SP_SYNC_FUNC	Meaning
0 (00b)	Rising edge
1 (01b)	Falling edge
2 (10b)	High
3 (11b)	Low

## 11.6 SP\_CON

<b>Bit</b>	<b>Name</b>
0	SP_COUNT_MODE
1	SP_COUNT_MODE
2	Reserved
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	SP_WRAP_COUNT
10	SP_ARM
11	SP_RST
12	Reserved
13	Reserved
14	SP_COUNT_UPDATED
15	SP_BUSY
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

**SP\_COUNT\_MODE**

R/W, SP\_CON[1..0], Aon-CXP, Axion-CL, Cyton-CXP

The mode that will be used for counting.

SP_COUNT_MODE	Meaning
0 (00b)	Current: on each update SP_COUNT will reflect the current event capture
1 (01b)	Accumulate: on each update SP_COUNT will reflect the Accumulation of all event captures since SP_CON.arm. Averaging can be handled by s/w in conjunction with the SP_STOP_EVENT_LIMIT
2 (10b)	Minimum: on each update SP_COUNT will reflect the Minimum of all event captures since SP_ARM
3 (11b)	Maximum: on each update SP_COUNT will reflect the Maximum of all event captures since SP_ARM

**SP\_WRAP\_COUNT**

R/W, SP\_CON[9], Aon-CXP, Axion-CL, Cyton-CXP

Controls how what the counter does when the maximum count is reached.

SP_WRAP_COUNT	Meaning
0	Counter saturates at max value
1	Counter wraps on overflow

**SP\_ARM**

WO, SP\_CON[10], Aon-CXP, Axion-CL, Cyton-CXP

Writing a 1 to this bit arms the counter. Counting will not start until both the SYNC\_EVENT and the START\_EVENT have occurred.

**SP\_RST**

WO, SP\_CON[11], Aon-CXP, Axion-CL, Cyton-CXP

Writing a 1 to this register resets the counter

**SP\_COUNT\_UPDATED**

RO, SP\_CON[14], Aon-CXP, Axion-CL, Cyton-CXP

SP\_COUNT will update at the end of each SP\_EVENTS\_STOP. If SP\_STOP\_EVENT\_LIMIT > 1, there will be more than 1 counter update. This bit signals is 1 when the count is updated. It is cleared if SP\_COUNT is read so that the user knows when SP\_COUNT represents fresh data



**SP\_BUSY**

RO, SP\_CON[15], Aon-CXP, Axion-CL, Cyton-CXP

The recorder is actively running. Reads back 1 when SP\_ARM is set. This reads back 0 when SP\_STOP\_EVENT\_LIMIT has been reached (the required number of events have been processed). A partial capture valid when SP\_COUNT\_UPDATED indicates an event window is complete, full capture valid when this bit reads back 0.

## 11.7 SP\_STAT

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

**SP\_COUNT**

RO, SP\_STAT[23..0], Aon-CXP, Axion-CL, Cyton-CXP

The number of target events that have occurred.

## 11.8 SP\_LIMIT

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

**SP\_STOP\_  
EVENT\_LIMIT**

R/W, SP\_LIMIT[23..0], Aon-CXP, Axion-CL, Cyton-CXP

The system will stop counting when this number of stop events have occurred.



# Specifications

## Chapter 12

### 12.1 Introduction

This chapter describes the general specifications of the Aon-CXP and Cyton-CXP family. The numerical values for the Cyton specifications are listed in Table 12-1 and the Aon specifications are in Table 12-2. If more information is available for a given specification there will be an entry in the column marked "Details".

Table 12-1 Cyton-CXP Specifications

Specifications	Value	Units	Details
PCIe Compatibility, slot type	x8 and x16	Slot size	Section 12.2
PCIe Compatibility, generation	Gen2 and 3		Section 12.2
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)	0.5	Amps	Section 12.3
Cyton-CXP4 Current (12 Volt)	0.62	Amps	Section 12.3
Cyton-CXP2 Current (3.3 Volt)	0.33	Amps	Section 12.3
Cyton-CXP2 Current (12 Volt)	0.60	Amps	Section 12.3
Temperature range	0 to 50	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	Per link
LVDS Drivers	SN65LVDS31D		
LVDS Receivers	SNLVDS3486		
TTL Drivers	SN74LVTH241		
TTL Receivers	SN74LVTH241		

Table 12-2 Aon-CXP Specifications

<b>Specifications</b>	<b>Value</b>	<b>Units</b>	<b>Details</b>
PCIe Compatibility, slot type	x4, x8 and x16	Slot size	Section 12.2
PCIe Compatibility, generation	Gen2 and 3		Section 12.2
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	
Aon-CXP Current (3.3 Volt)	6.5	Amps	Section 12.3
Aon-CXP Current (12 Volt)	0.14	Amps	Section 12.3
Temperature range	0 to 50	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	Per link
LVDS Drivers	SN65LVDS31D		
LVDS Receivers	SNLVDS3486		
TTL Drivers	SN74LVTH241		
TTL Receivers	SN74LVTH241		



## 12.2 PCI Express Compatibility

The Cyton-CXP is a PCIe x8 Gen 2 board. However, it will work in any PCIe slot that it fits into. This means it will work in x8 and x16 slots, however, it will also work in x1 and x4 slots if these slots are mechanically compatible with an x8 board, though performance will be greatly degraded. Similarly, the Cyton-CXP will work in Gen 1, Gen 2 and Gen 3 slot, however, performance will be degraded in a Gen 1 slot. Further, there is no performance gained by putting the Cyton-CXP in a Gen 3 slot, performance will be the same as a Gen 2 slot.

*Note: For best DMA performance, put the Cyton-CXP in a PCIe x8 or x16 Gen 2 or Gen 3 slot on a high quality motherboard.*

The Aon-CXP is a PCIe x2 Gen 2 board. It will work in any slot that you can plug it into physically. However, in order to support a CXP camera running at 6.25 Gb/S, you must put it in a Gen 2 or Gen 3 slot that is x2, x4, x8 or x16. When working with slower cameras, it is possible that you have sufficient performance in an x1 Gen 2 or Gen 3 slot.

## 12.3 Power Requirements

The power requirements listed in Table 12-1 are the requirements of the board's circuitry only. In addition, the Cyton-CXP and Aon-CXP can provide up to 13 watts of power to each CXP link. In the case of the Cyton, 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 used 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 13.6 for more information on this jumper.

*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 watts on two links is connected, then auxiliary power must be used.*

The Aon 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 13.7 for more information.

# Mechanical

## Chapter 13

### 13.1 Introduction

This chapter describes the mechanical characteristics of the Cyton-CXP. This includes description of all of the connectors on the board

and pin-outs for these connectors.

The mechanical layout of the Cyton-CXP4 board is shown in Figure 13-1 and the Cyton-CXP2 is shown in Figure 13-2. The Aon-CXP is shown in Figure 13-3

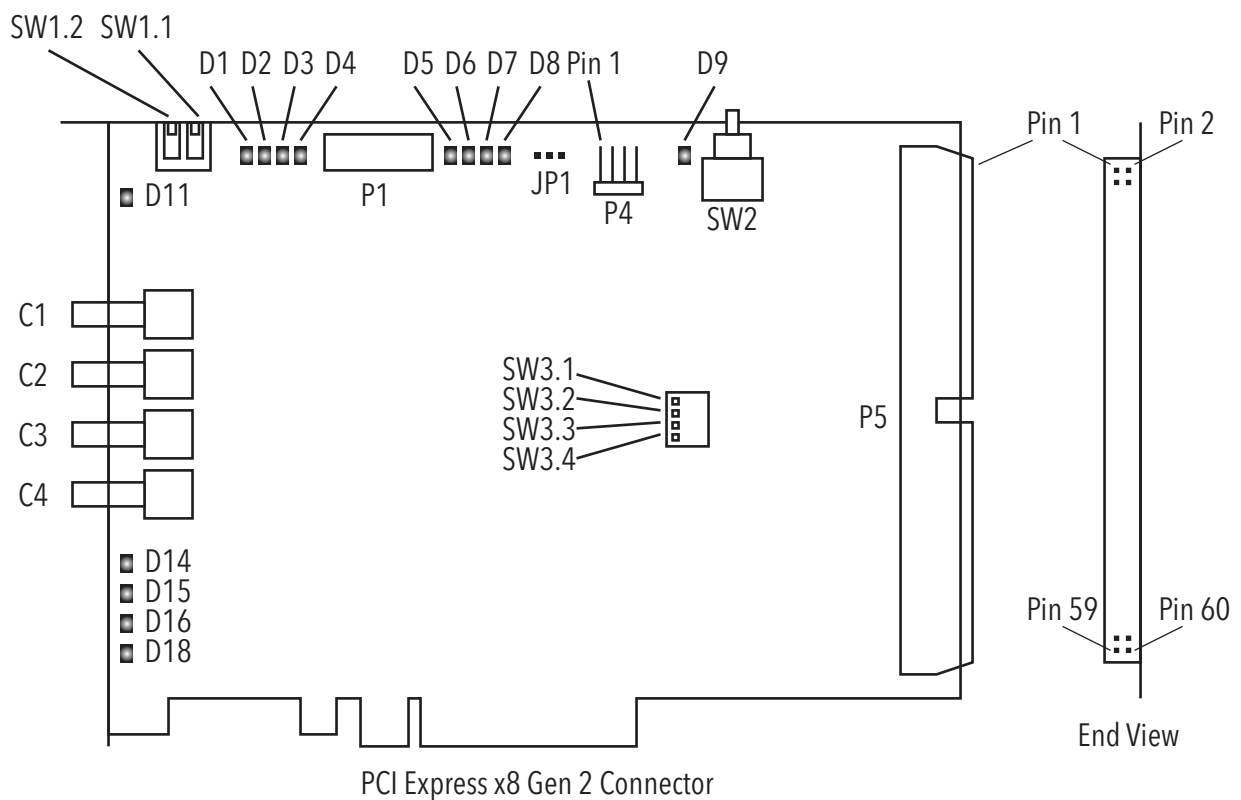


Figure 13-1 Cyton-CXP4 Board Layout

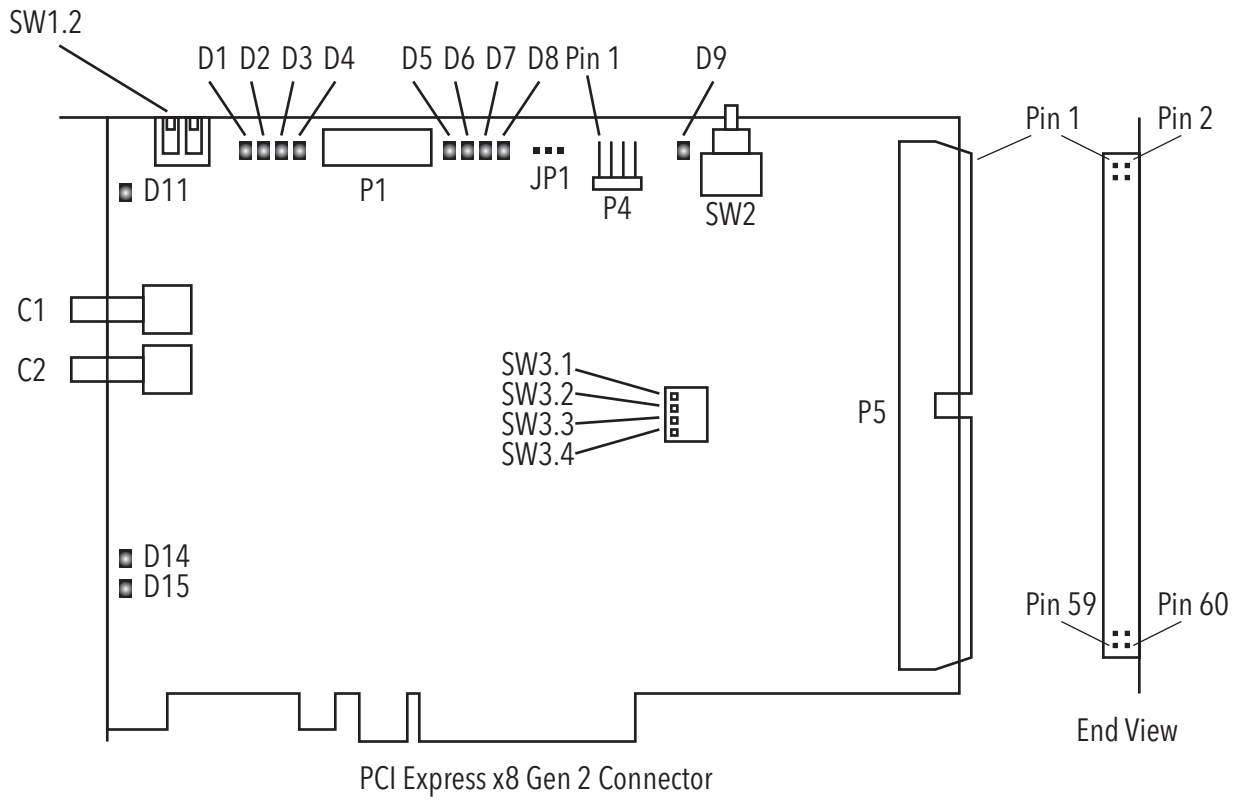


Figure 13-2 Cyton-CXP2 Board Layout

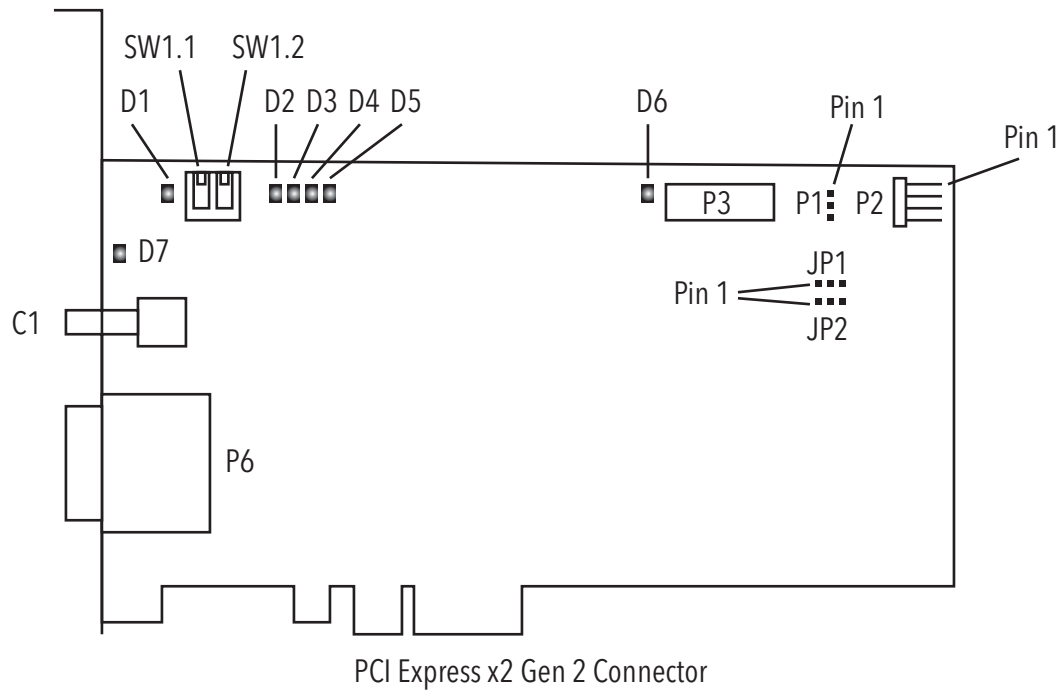


Figure 13-3 Aon-CXP Board Layout

## 13.2 The Cyton-CXP Connectors

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 13-1 shows the locations of these connectors. The following sections show the details of each of these connectors.

### 13.2.1 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 13-1 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 13-1 CXP Connector Configuration, Cyton-CXP4**

<b>Camera(s)</b>	<b>C1</b>	<b>C2</b>	<b>C3</b>	<b>C4</b>
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

## 13.3 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 13-3 shows the locations of these connectors. The following sections show the details of each of these connectors.

### 13.3.1 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.

## 13.4 The Cyton Switches

There is one piano-type switch block, SW1, on the Cyton-CXP 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 13-2 below shows the switch settings and the corresponding value in the SW bitfield.

**Table 13-2 Switch S1 Setting**

<b>SW1.1</b>	<b>SW1.2</b>	<b>SW register</b>
down	down	0
down	up	1
up	down	2
up	up	3

There is one micro switch block, SW3, on the Cyton-CXP with four switches. These are used to control the flash bank that the system boots from.

*Note: Do not change these switches unless instructed by BitFlow support.*

See Table 13-3 below which shows the switch settings and the corresponding firmware bank.

**Table 13-3 Switch S3 Setting**

<b>SW3.4</b>	<b>SW3.2</b>	<b>SW 3.2</b>	<b>SW 3.1</b>	<b>FW Bank</b>
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 13-3 Switch S3 Setting

<b>SW3.4</b>	<b>SW3.2</b>	<b>SW 3.2</b>	<b>SW 3.1</b>	<b>FW Bank</b>
on	on	off	off	Reserved
on	on	off	on	Reserved
on	on	on	off	Reserved
on	on	on	on	Reserved

## 13.5 The Aon Switches

There is one piano-type switch block, SW1, on the Aon-CXP 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 13-4 below shows the switch settings and the corresponding value in the SW bitfield.

Table 13-4 Aon Switch S1 Setting

<b>SW1.1</b>	<b>SW1.2</b>	<b>SW register</b>
down	down	0
down	up	1
up	down	2
up	up	3

## 13.6 The Cyton Jumpers

There is one user configurable jumper on the Cyton-CXP, it controls the source of the power that is provided to the CoaXPress camera(s) connected to the Cyton (PoCXP power). Table 13-5 shows the two settings.

Table 13-5 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.*

## 13.7 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 13-6 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 13-3 for the location of the Jumper P1.*

Table 13-6 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.

## 13.8 The Cyton LEDs

The Cyton-CXP has 14 LEDs. Table 13-7 describes the function of these LEDs.

*Note: The meanings of the LEDs where color is shown a "Various" is described in Table 13-8 and Table 13-9.*

Table 13-7 Cyton LEDs

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

### 13.8.1 CXP Downlink LED Meaning

The LEDs D14, D15, D16 and D18 indicate the status of the CXP downlink. The meaning of each color and blink mode is described in Table 13-8.

Table 13-8 Downlink 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	Packet has CRC error

### 13.8.2 CXP Uplink LED Meaning

The LED D11 indicate the status of the CXP uplink. The meaning of each color and blink mode is described in Table 13-9.

Table 13-9 Uplink LED meaning

Color	State	Meaning
Blue	Blinking	TBD
Blue	Steady	TBD
Green	Steady	TBD
Green	Blinking	TBD
Red	Blinking	TBD

## 13.9 The Aon LEDs

The Aon-CXP has 7 LEDs. Table 13-10 describes the function of these LEDs.

*Note: The meanings of the LEDs where color is shown a "Various" is described in Table 13-8.*

Table 13-10 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

## 13.10 Button

The Cyton-CXP 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.



## 13.11 The Auxiliary Power Connector (Cyton P4, Aon P2 )

The Aon and Cyton 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 it may be desirable or necessary to use the PC's power supply to supply power. On the Cyton, if all four CXP links are taking maximum power, then the PCIe bus can not provide sufficient power and the PC's power supply must be used. On the Aon, 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.

The Auxiliary power connector accepts 12 Volts from the PC's power supply. There is a jumper (JP1 on the Cyton, P1 on the Aon) that must be in set to the correct position to route power from this connector to the CoaXPress connectors.

It is possible to provide power to cameras that require more power than can be provided by the PCIe bus, the Cyton-CXP has a connector, P4, which can take auxiliary power from the PC's power supply. The pin out for this connector is shown in Table 13-11.

*Note: The Auxiliary Power Connector is only used for powering the connected CXP camera, current from this connector is not used to power the components on the Cyton/Aon.*

*Note: The Auxiliary Power 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 13-11 Auxiliary Power Connector**

<b>Pin</b>	<b>Voltage</b>
1	NC
2	GND
3	GND
4	12 Volts

## 13.12 The BitBox Connector (Cyton P1, Aon P3)

This connector is for connection BitFlow external I/O box, called the BitBox. Contact BitFlow for more information on the BitBox.

### 13.13 I/O Connector Pinout for the Cyton-CXP P5

The pin-out for the I/O Connector for the Cyton-CXP is illustrated in the Table 13-12.

*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 13-12 I/O Connector for the Cyton-CXP

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
32	Out	VFG3_CC3+	LVDS

Table 13-12 I/O Connector for the Cyton-CXP

Pin	I/O	Signal	Comment
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	

## 13.14 I/O Connector Pinout for the Aon-CXP P6

The pin-out for the I/O Connector for the Aon-CXP is illustrated in the Table 13-13.

Table 13-13 I/O Connector for the Aon-CXP

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

0\_COM\_RCV\_DATA CYT-10-19  
 0\_COM\_RCV\_FIFO\_CLR CYT-10-17  
 0\_COM\_RCV\_FIFO\_CNT CYT-10-17  
 0\_COM\_RCV\_FIFO\_SIZE CYT-10-11  
 0\_COM\_SEND\_DATA CYT-10-15  
 0\_COM\_SEND\_FIFO\_CLR CYT-10-13  
 0\_COM\_SEND\_FIFO\_CNT CYT-10-13  
 0\_COM\_SEND\_FIFO\_SIZE CYT-10-11  
 0\_COM\_SEND\_GO CYT-10-13  
 0\_CRC\_ERR\_CNT CYT-10-35  
 0\_CTL\_ACK\_RCVD CYT-10-23  
 0\_CTL\_ACK\_RCVD\_CM CYT-10-30  
 0\_CTL\_ACK\_RCVD\_M CYT-10-27  
 0\_CTL\_REQ\_FIFO\_OVF CYT-10-23  
 0\_CTL\_REQ\_FIFO\_OVF\_CM CYT-10-30  
 0\_CTL\_REQ\_FIFO\_OVF\_M CYT-10-27  
 0\_CTL\_RSP\_FIFO\_OVF CYT-10-23  
 0\_CTL\_RSP\_FIFO\_OVF\_CM CYT-10-30  
 0\_CTL\_RSP\_FIFO\_OVF\_M CYT-10-27  
 0\_CXP\_TRIG\_ACK\_CNT CYT-10-37  
 0\_CXP\_TRIG\_FALL\_SENT CYT-10-37  
 0\_CXP\_TRIG\_STATE CYT-10-37  
 0\_DOWN\_TRIG\_RCVD CYT-10-23, CYT-10-27  
 0\_DOWN\_TRIG\_RCVD\_WP CYT-10-30  
 0\_DT\_CLR\_CNT CYT-10-39  
 0\_DT\_FALL\_DISABLE CYT-10-39  
 0\_DT\_FALL\_RCVD\_COUNT CYT-10-39  
 0\_DT\_RISE\_DISABLE CYT-10-39  
 0\_DT\_RISE\_RCVD\_COUNT CYT-10-39  
 0\_DT\_STATE CYT-10-39  
 0\_DT\_TRIG\_DELAY CYT-10-39  
 0\_GPIO\_ACK\_RCVD CYT-10-23  
 0\_GPIO\_ACK\_RCVD\_CM CYT-10-30  
 0\_GPIO\_ACK\_RCVD\_M CYT-10-27  
 0\_GPIO\_RCVD CYT-10-23  
 0\_GPIO\_RCVD\_CM CYT-10-30  
 0\_GPIO\_RCVD\_M CYT-10-27  
 0\_IOACK\_NOMATCH CYT-10-24  
 0\_IOACK\_NOMATCH\_CM CYT-10-31  
 0\_IOACK\_NOMATCH\_M CYT-10-28  
 0\_IOACK\_NOMATCH2 CYT-10-24  
 0\_IOACK\_NOMATCH2\_CM CYT-10-31  
 0\_IOACK\_NOMATCH2\_M CYT-10-28  
 0\_IOACK\_UNEXPECTED\_INT CYT-10-24  
 0\_IOACK\_UNEXPECTED\_INT\_CM CYT-10-31  
 0\_IOACK\_UNEXPECTED\_INT\_M CYT-10-28  
 0\_IOACK\_UNKNOWN\_TYPE CYT-10-24  
 0\_IOACK\_UNKNOWN\_TYPE\_CM CYT-10-31  
 0\_IOACK\_UNKNOWN\_TYPE\_M CYT-10-28  
 0\_LINK\_INT\_DEST CYT-10-21  
 0\_LINK\_SPEED CYT-10-41  
 0\_LOST\_ALIGN\_CNT CYT-10-41  
 0\_OVER\_CURRENT CYT-10-23  
 0\_OVER\_CURRENT\_CM CYT-10-30  
 0\_OVER\_CURRENT\_M CYT-10-27  
 0\_PKT\_DROP\_CNT CYT-10-35  
 0\_PKT\_GNT\_CNT CYT-10-33  
 0\_PKT\_RCVD\_CNT CYT-10-33  
 0\_POCXP\_24V\_OK CYT-10-4, CYT-10-48, CYT-10-90, CYT-10-132  
 0\_POCXP\_CAM\_IS\_POCXP CYT-10-3  
 0\_POCXP\_CURRENT CYT-10-5  
 0\_POCXP\_CURRENT\_LATCH CYT-10-4  
 0\_POCXP\_EN\_24V\_REG CYT-10-3  
 0\_POCXP\_EN\_CAM\_SENSE CYT-10-3  
 0\_POCXP\_EN\_POWER CYT-10-3  
 0\_POCXP\_OPEN\_DETECTED CYT-10-3  
 0\_POCXP\_OVER\_DETECTED CYT-10-3  
 0\_POCXP\_OVER\_LATCH CYT-10-3  
 0\_POCXP\_OVER\_TIMER CYT-10-7  
 0\_POCXP\_SHORT\_DETECTED CYT-10-3  
 0\_POCXP\_UNDER\_DETECTED CYT-10-4  
 0\_POCXP\_UNDER\_LATCH CYT-10-4  
 0\_POCXP\_UNDER\_TIMER CYT-10-9  
 0\_RAW\_DATA\_MODE CYT-10-43  
 0\_REMOVE\_IDLES CYT-10-43  
 0\_RESYNC\_TOG CYT-10-41  
 0\_RESYNC\_TOG\_MSTR CYT-10-41  
 0\_SERDES\_ALIGNED CYT-10-41  
 0\_SERDES\_ERROR\_CODE CYT-10-45  
 0\_SERDES\_LOST\_ALIGN CYT-10-25  
 0\_SERDES\_STATE CYT-10-41  
 0\_STRM\_BAD\_CRC CYT-10-24  
 0\_STRM\_BAD\_CRC\_CM CYT-10-31  
 0\_STRM\_BAD\_CRC\_M CYT-10-28  
 0\_STRM\_CORNER CYT-10-25  
 0\_STRM\_NOT\_ENOUGH\_DAT CYT-10-24  
 0\_STRM\_NOT\_ENOUGH\_DAT\_CM CYT-10-31  
 0\_STRM\_NOT\_ENOUGH\_DAT\_M CYT-10-28  
 0\_STRM\_OVERFLOW CYT-10-24

0\_STRM\_OVERFLOW\_CM CYT-10-31  
 0\_STRM\_OVERFLOW\_M CYT-10-28  
 0\_STRM\_PKT\_DROP CYT-10-24  
 0\_STRM\_PKT\_DROP\_CM CYT-10-31  
 0\_STRM\_PKT\_DROP\_M CYT-10-28  
 0\_STRM\_TOO\_CMUCH\_DAT\_CM CYT-10-31  
 0\_STRM\_TOO\_MUCH\_DAT CYT-10-24  
 0\_STRM\_TOO\_MUCH\_DAT\_M CYT-10-28  
 0\_TRIG\_ACK\_RCVD CYT-10-23  
 0\_TRIG\_ACK\_RCVD\_CM CYT-10-30  
 0\_TRIG\_ACK\_RCVD\_M CYT-10-27  
 0\_TRIG\_NOMATCH CYT-10-24  
 0\_TRIG\_NOMATCH\_CM CYT-10-31  
 0\_TRIG\_NOMATCH\_M CYT-10-28  
 0\_TRIG\_RCVD CYT-10-23  
 0\_TRIG\_RCVD\_CM CYT-10-30  
 0\_TRIG\_RCVD\_M CYT-10-27  
 0\_UNDER\_CURRENT CYT-10-23  
 0\_UNDER\_CURRENT\_CM CYT-10-30  
 0\_UNDER\_CURRENT\_M CYT-10-27  
 1\_COM\_RCV\_DATA CYT-10-62  
 1\_COM\_RCV\_FIFO\_CLR CYT-10-60  
 1\_COM\_RCV\_FIFO\_CNT CYT-10-60  
 1\_COM\_RCV\_FIFO\_SIZE CYT-10-54  
 1\_COM\_SEND\_DATA CYT-10-58  
 1\_COM\_SEND\_FIFO\_CLR CYT-10-56  
 1\_COM\_SEND\_FIFO\_CNT CYT-10-56  
 1\_COM\_SEND\_FIFO\_SIZE CYT-10-54  
 1\_COM\_SEND\_GO CYT-10-56  
 1\_CRC\_ERR\_CNT CYT-10-77  
 1\_CTL\_ACK\_RCVD CYT-10-66  
 1\_CTL\_ACK\_RCVD\_CM CYT-10-72  
 1\_CTL\_ACK\_RCVD\_M CYT-10-69  
 1\_CTL\_REQ\_FIFO\_OVF CYT-10-66  
 1\_CTL\_REQ\_FIFO\_OVF\_CM CYT-10-72  
 1\_CTL\_REQ\_FIFO\_OVF\_M CYT-10-69  
 1\_CTL\_RSP\_FIFO\_OVF CYT-10-66  
 1\_CTL\_RSP\_FIFO\_OVF\_CM CYT-10-72  
 1\_CTL\_RSP\_FIFO\_OVF\_M CYT-10-69  
 1\_CXP\_TRIG\_ACK\_CNT CYT-10-79  
 1\_CXP\_TRIG\_FALL\_SENT CYT-10-79  
 1\_CXP\_TRIG\_STATE CYT-10-79  
 1\_DOWN\_TRIG\_RCVD\_CM CYT-10-72  
 1\_DOWN\_TRIG\_RCVD\_M CYT-10-69  
 1\_DT\_CLR\_CNT CYT-10-81  
 1\_DT\_FALL\_DISABLE CYT-10-81  
 1\_DT\_FALL\_RCVD\_COUNT CYT-10-81  
 1\_DT\_RISE\_DISABLE CYT-10-81  
 1\_DT\_RISE\_RCVD\_COUNT CYT-10-81  
 1\_DT\_STATE CYT-10-81  
 1\_DT\_TRIG\_DELAY CYT-10-81  
 1\_GDOWN\_TRIG\_RCVD CYT-10-66  
 1\_GPIO\_ACK\_RCVD CYT-10-66  
 1\_GPIO\_ACK\_RCVD\_CM CYT-10-72  
 1\_GPIO\_ACK\_RCVD\_M CYT-10-69  
 1\_GPIO\_RCVD CYT-10-66  
 1\_GPIO\_RCVD\_CM CYT-10-72  
 1\_GPIO\_RCVD\_M CYT-10-69  
 1\_IOACK\_NOMATCH CYT-10-67  
 1\_IOACK\_NOMATCH\_CM CYT-10-73  
 1\_IOACK\_NOMATCH\_M CYT-10-70  
 1\_IOACK\_NOMATCH2 CYT-10-67  
 1\_IOACK\_NOMATCH2\_CM CYT-10-73  
 1\_IOACK\_NOMATCH2\_M CYT-10-70  
 1\_IOACK\_UNEXPECTED\_INT CYT-10-67  
 1\_IOACK\_UNEXPECTED\_INT\_CM CYT-10-73  
 1\_IOACK\_UNEXPECTED\_INT\_M CYT-10-70  
 1\_IOACK\_UNKNOWN\_TYPE CYT-10-67  
 1\_IOACK\_UNKNOWN\_TYPE\_CM CYT-10-73  
 1\_IOACK\_UNKNOWN\_TYPE\_M CYT-10-70  
 1\_LINK\_INT\_DEST CYT-10-64  
 1\_LINK\_SPEED CYT-10-83  
 1\_LOST\_ALIGN\_CNT CYT-10-83  
 1\_OVER\_CURRENT CYT-10-66  
 1\_OVER\_CURRENT\_CM CYT-10-72  
 1\_OVER\_CURRENT\_M CYT-10-69  
 1\_PKT\_DROP\_CNT CYT-10-77  
 1\_PKT\_GNT\_CNT CYT-10-75  
 1\_PKT\_RCVD\_CNT CYT-10-75  
 1\_POCXP\_24V\_OK CYT-10-48  
 1\_POCXP\_CAM\_IS\_POCXP CYT-10-47  
 1\_POCXP\_CURRENT CYT-10-48  
 1\_POCXP\_CURRENT\_LATCH CYT-10-48  
 1\_POCXP\_EN\_24V\_REG CYT-10-47  
 1\_POCXP\_EN\_CAM\_SENSE CYT-10-47  
 1\_POCXP\_EN\_POWER CYT-10-47  
 1\_POCXP\_OPEN\_DETECTED CYT-10-47  
 1\_POCXP\_OVER\_DETECTED CYT-10-47  
 1\_POCXP\_OVER\_LATCH CYT-10-47  
 1\_POCXP\_OVER\_TIMER CYT-10-50  
 1\_POCXP\_SHORT\_DETECTED CYT-10-47  
 1\_POCXP\_UNDER\_DETECTED CYT-10-47  
 1\_POCXP\_UNDER\_LATCH CYT-10-47  
 1\_POCXP\_UNDER\_TIMER CYT-10-52  
 1\_RAW\_DATA\_MODE CYT-10-85  
 1\_REMOVE\_IDLES CYT-10-85  
 1\_RESYNC\_TOG CYT-10-83  
 1\_RESYNC\_TOG\_MSTR CYT-10-83  
 1\_SERDES\_ALIGNED CYT-10-83  
 1\_SERDES\_ERROR\_CODE CYT-10-87



1\_SERDES\_STATE CYT-10-83  
 1\_STRM\_BAD\_CRC CYT-10-67  
 1\_STRM\_BAD\_CRC\_CM CYT-10-73  
 1\_STRM\_BAD\_CRC\_M CYT-10-70  
 1\_STRM\_NOT\_ENOUGH\_DAT CYT-10-67  
 1\_STRM\_NOT\_ENOUGH\_DAT\_CM CYT-10-73  
 1\_STRM\_NOT\_ENOUGH\_DAT\_M CYT-10-70  
 1\_STRM\_OVERFLOW CYT-10-67  
 1\_STRM\_OVERFLOW\_CM CYT-10-73  
 1\_STRM\_OVERFLOW\_M CYT-10-70  
 1\_STRM\_PKT\_DROP CYT-10-67  
 1\_STRM\_PKT\_DROP\_CM CYT-10-73  
 1\_STRM\_PKT\_DROP\_M CYT-10-70  
 1\_STRM\_TOO\_CMUCH\_DAT\_CM CYT-10-73  
 1\_STRM\_TOO\_MUCH\_DAT CYT-10-67  
 1\_STRM\_TOO\_MUCH\_DAT\_M CYT-10-70  
 1\_TRIG\_ACK\_RCVD CYT-10-66  
 1\_TRIG\_ACK\_RCVD\_CM CYT-10-72  
 1\_TRIG\_ACK\_RCVD\_M CYT-10-69  
 1\_TRIG\_NOMATCH CYT-10-67  
 1\_TRIG\_NOMATCH\_CM CYT-10-73  
 1\_TRIG\_NOMATCH\_M CYT-10-70  
 1\_TRIG\_RCVD CYT-10-66  
 1\_TRIG\_RCVD\_CM CYT-10-72  
 1\_TRIG\_RCVD\_M CYT-10-69  
 1\_UNDER\_CURRENT CYT-10-66  
 1\_UNDER\_CURRENT\_CM CYT-10-72  
 1\_UNDER\_CURRENT\_M CYT-10-69  
 2\_COM\_RCV\_DATA CYT-10-104  
 2\_COM\_RCV\_FIFO\_CLR CYT-10-102  
 2\_COM\_RCV\_FIFO\_CNT CYT-10-102  
 2\_COM\_RCV\_FIFO\_SIZE CYT-10-96  
 2\_COM\_SEND\_DATA CYT-10-100  
 2\_COM\_SEND\_FIFO\_CLR CYT-10-98  
 2\_COM\_SEND\_FIFO\_CNT CYT-10-98  
 2\_COM\_SEND\_FIFO\_SIZE CYT-10-96  
 2\_COM\_SEND\_GO CYT-10-98  
 2\_CRC\_ERR\_CNT CYT-10-119  
 2\_CTL\_ACK\_RCVD CYT-10-108  
 2\_CTL\_ACK\_RCVD\_CM CYT-10-114  
 2\_CTL\_ACK\_RCVD\_M CYT-10-111  
 2\_CTL\_REQ\_FIFO\_OVF CYT-10-108  
 2\_CTL\_REQ\_FIFO\_OVF\_CM CYT-10-114  
 2\_CTL\_REQ\_FIFO\_OVF\_M CYT-10-111  
 2\_CTL\_RSP\_FIFO\_OVF CYT-10-108  
 2\_CTL\_RSP\_FIFO\_OVF\_CM CYT-10-114  
 2\_CTL\_RSP\_FIFO\_OVF\_M CYT-10-111  
 2\_CXP\_TRIG\_ACK\_CNT CYT-10-121  
 2\_CXP\_TRIG\_FALL\_SENT CYT-10-121  
 2\_CXP\_TRIG\_STATE CYT-10-121  
 2\_DOWN\_TRIG\_RCVD CYT-10-108  
 2\_DOWN\_TRIG\_RCVD\_CM CYT-10-114  
 2\_DOWN\_TRIG\_RCVD\_M CYT-10-111  
 2\_DT\_CLR\_CNT CYT-10-123  
 2\_DT\_FALL\_DISABLE CYT-10-123  
 2\_DT\_FALL\_RCVD\_COUNT CYT-10-123  
 2\_DT\_RISE\_DISABLE CYT-10-123  
 2\_DT\_RISE\_RCVD\_COUNT CYT-10-123  
 2\_DT\_STATE CYT-10-123  
 2\_DT\_TRIG\_DELAY CYT-10-123  
 2\_GPIO\_ACK\_RCVD CYT-10-108  
 2\_GPIO\_ACK\_RCVD\_CM CYT-10-114  
 2\_GPIO\_ACK\_RCVD\_M CYT-10-111  
 2\_GPIO\_RCVD CYT-10-108  
 2\_GPIO\_RCVD\_CM CYT-10-114  
 2\_GPIO\_RCVD\_M CYT-10-111  
 2\_IOACK\_NOMATCH CYT-10-109  
 2\_IOACK\_NOMATCH\_CM CYT-10-115  
 2\_IOACK\_NOMATCH\_M CYT-10-112  
 2\_IOACK\_NOMATCH2 CYT-10-109  
 2\_IOACK\_NOMATCH2\_CM CYT-10-115  
 2\_IOACK\_NOMATCH2\_M CYT-10-112  
 2\_IOACK\_UNEXPECTED\_INT CYT-10-109  
 2\_IOACK\_UNEXPECTED\_INT\_CM CYT-10-115  
 2\_IOACK\_UNEXPECTED\_INT\_M CYT-10-112  
 2\_IOACK\_UNKNOWN\_TYPE CYT-10-109  
 2\_IOACK\_UNKNOWN\_TYPE\_CM CYT-10-115  
 2\_IOACK\_UNKNOWN\_TYPE\_M CYT-10-112  
 2\_LINK\_INT\_DEST CYT-10-106  
 2\_LINK\_SPEED CYT-10-125  
 2\_LOST\_ALIGN\_CNT CYT-10-125  
 2\_OVER\_CURRENT CYT-10-108  
 2\_OVER\_CURRENT\_CM CYT-10-114  
 2\_OVER\_CURRENT\_M CYT-10-111  
 2\_PKT\_DROP\_CNT CYT-10-119  
 2\_PKT\_GNT\_CNT CYT-10-117  
 2\_PKT\_RCVD\_CNT CYT-10-117  
 2\_POCPX\_24V\_OK CYT-10-90  
 2\_POCPX\_CAM\_IS\_POCPX CYT-10-89  
 2\_POCPX\_CURRENT CYT-10-90  
 2\_POCPX\_CURRENT\_LATCH CYT-10-90  
 2\_POCPX\_EN\_24V\_REG CYT-10-89  
 2\_POCPX\_EN\_CAM\_SENSE CYT-10-89  
 2\_POCPX\_EN\_POWER CYT-10-89  
 2\_POCPX\_OPEN\_DETECTED CYT-10-89  
 2\_POCPX\_OVER\_DETECTED CYT-10-89  
 2\_POCPX\_OVER\_LATCH CYT-10-89  
 2\_POCPX\_OVER\_TIMER CYT-10-92  
 2\_POCPX\_SHORT\_DETECTED CYT-10-89  
 2\_POCPX\_UNDER\_DETECTED CYT-10-89

2\_POCPX\_UNDER\_LATCH CYT-10-89  
 2\_POCPX\_UNDER\_TIMER CYT-10-94  
 2\_RAW\_DATA\_MODE CYT-10-127  
 2\_REMOVE\_IDLES CYT-10-127  
 2\_RESYNC\_TOG CYT-10-125  
 2\_RESYNC\_TOG\_MSTR CYT-10-125  
 2\_SERDES\_ALIGNED CYT-10-125  
 2\_SERDES\_ERROR\_CODE CYT-10-129  
 2\_SERDES\_STATE CYT-10-125  
 2\_STRM\_BAD\_CRC CYT-10-109  
 2\_STRM\_BAD\_CRC\_CM CYT-10-115  
 2\_STRM\_BAD\_CRC\_M CYT-10-112  
 2\_STRM\_NOT\_ENOUGH\_DAT CYT-10-109  
 2\_STRM\_NOT\_ENOUGH\_DAT\_CM CYT-10-115  
 2\_STRM\_NOT\_ENOUGH\_DAT\_M CYT-10-112  
 2\_STRM\_OVERFLOW CYT-10-109  
 2\_STRM\_OVERFLOW\_CM CYT-10-115  
 2\_STRM\_OVERFLOW\_M CYT-10-112  
 2\_STRM\_PKT\_DROP CYT-10-109  
 2\_STRM\_PKT\_DROP\_CM CYT-10-115  
 2\_STRM\_PKT\_DROP\_M CYT-10-112  
 2\_STRM\_TOO\_CMUCH\_DAT\_CM CYT-10-115  
 2\_STRM\_TOO\_MUCH\_DAT CYT-10-109  
 2\_STRM\_TOO\_MUCH\_DAT\_M CYT-10-112  
 2\_TRIG\_ACK\_RCVD CYT-10-108  
 2\_TRIG\_ACK\_RCVD\_CM CYT-10-114  
 2\_TRIG\_ACK\_RCVD\_M CYT-10-111  
 2\_TRIG\_NOMATCH CYT-10-109  
 2\_TRIG\_NOMATCH\_CM CYT-10-115  
 2\_TRIG\_NOMATCH\_M CYT-10-112  
 2\_TRIG\_RCVD CYT-10-108  
 2\_TRIG\_RCVD\_CM CYT-10-114  
 2\_TRIG\_RCVD\_M CYT-10-111  
 2\_UNDER\_CURRENT CYT-10-108  
 2\_UNDER\_CURRENT\_CM CYT-10-114  
 2\_UNDER\_CURRENT\_M CYT-10-111  
 3\_COM\_RCV\_DATA CYT-10-146  
 3\_COM\_RCV\_FIFO\_CLR CYT-10-144  
 3\_COM\_RCV\_FIFO\_CNT CYT-10-144  
 3\_COM\_RCV\_FIFO\_SIZE CYT-10-138  
 3\_COM\_SEND\_DATA CYT-10-142  
 3\_COM\_SEND\_FIFO\_CLR CYT-10-140  
 3\_COM\_SEND\_FIFO\_CNT CYT-10-140  
 3\_COM\_SEND\_FIFO\_SIZE CYT-10-138  
 3\_COM\_SEND\_GO CYT-10-140  
 3\_CRC\_ERR\_CNT CYT-10-161  
 3\_CTL\_ACK\_RCVD CYT-10-150  
 3\_CTL\_ACK\_RCVD\_CM CYT-10-156  
 3\_CTL\_ACK\_RCVD\_M CYT-10-153  
 3\_CTL\_REQ\_FIFO\_OVF CYT-10-150  
 3\_CTL\_REQ\_FIFO\_OVF\_CM CYT-10-156  
 3\_CTL\_REQ\_FIFO\_OVF\_M CYT-10-153  
 3\_CTL\_RSP\_FIFO\_OVF CYT-10-150  
 3\_CTL\_RSP\_FIFO\_OVF\_CM CYT-10-156  
 3\_CTL\_RSP\_FIFO\_OVF\_M CYT-10-153  
 3\_CXP\_TRIG\_ACK\_CNT CYT-10-163, CYT-10-165  
 3\_CXP\_TRIG\_FALL\_SENT CYT-10-163  
 3\_CXP\_TRIG\_STATE CYT-10-163, CYT-10-165  
 3\_DOWN\_TRIG\_RCVD CYT-10-150  
 3\_DOWN\_TRIG\_RCVD\_CM CYT-10-156  
 3\_DOWN\_TRIG\_RCVD\_M CYT-10-153  
 3\_DT\_CLR\_CNT CYT-10-167  
 3\_DT\_FALL\_DISABLE CYT-10-167  
 3\_DT\_FALL\_RCVD\_COUNT CYT-10-167  
 3\_DT\_RISE\_DISABLE CYT-10-167  
 3\_DT\_RISE\_RCVD\_COUNT CYT-10-167  
 3\_DT\_STATE CYT-10-167  
 3\_DT\_TRIG\_DELAY CYT-10-167  
 3\_GPIO\_ACK\_RCVD CYT-10-150  
 3\_GPIO\_ACK\_RCVD\_CM CYT-10-156  
 3\_GPIO\_ACK\_RCVD\_M CYT-10-153  
 3\_GPIO\_RCVD CYT-10-150  
 3\_GPIO\_RCVD\_CM CYT-10-156  
 3\_GPIO\_RCVD\_M CYT-10-153  
 3\_IOACK\_NOMATCH CYT-10-151  
 3\_IOACK\_NOMATCH\_CM CYT-10-157  
 3\_IOACK\_NOMATCH\_M CYT-10-154  
 3\_IOACK\_NOMATCH2 CYT-10-151  
 3\_IOACK\_NOMATCH2\_CM CYT-10-157  
 3\_IOACK\_NOMATCH2\_M CYT-10-154  
 3\_IOACK\_UNEXPECTED\_INT CYT-10-151  
 3\_IOACK\_UNEXPECTED\_INT\_CM CYT-10-157  
 3\_IOACK\_UNEXPECTED\_INT\_M CYT-10-154  
 3\_IOACK\_UNKNOWN\_TYPE CYT-10-151  
 3\_IOACK\_UNKNOWN\_TYPE\_CM CYT-10-157  
 3\_IOACK\_UNKNOWN\_TYPE\_M CYT-10-154  
 3\_LINK\_INT\_DEST CYT-10-148  
 3\_LINK\_SPEED CYT-10-169  
 3\_LOST\_ALIGN\_CNT CYT-10-169  
 3\_OVER\_CURRENT CYT-10-150  
 3\_OVER\_CURRENT\_CM CYT-10-156  
 3\_OVER\_CURRENT\_M CYT-10-153  
 3\_PKT\_DROP\_CNT CYT-10-161  
 3\_PKT\_GNT\_CNT CYT-10-159  
 3\_PKT\_RCVD\_CNT CYT-10-159  
 3\_POCPX\_24V\_OK CYT-10-132  
 3\_POCPX\_CAM\_IS\_POCPX CYT-10-131  
 3\_POCPX\_CURRENT CYT-10-132

3\_POCPX\_CURRENT\_LATCH CYT-10-132  
 3\_POCPX\_EN\_24V\_REG CYT-10-131  
 3\_POCPX\_EN\_CAM\_SENSE CYT-10-131  
 3\_POCPX\_EN\_POWER CYT-10-131  
 3\_POCPX\_OPEN\_DETECTED CYT-10-131  
 3\_POCPX\_OVER\_DETECTED CYT-10-131  
 3\_POCPX\_OVER\_LATCH CYT-10-131  
 3\_POCPX\_OVER\_TIMER CYT-10-134  
 3\_POCPX\_SHORT\_DETECTED CYT-10-131  
 3\_POCPX\_UNDER\_DETECTED CYT-10-131  
 3\_POCPX\_UNDER\_LATCH CYT-10-131  
 3\_POCPX\_UNDER\_TIMER CYT-10-136  
 3\_RAW\_DATA\_MODE CYT-10-171  
 3\_REMOVE\_IDLES CYT-10-171  
 3\_RESYNC\_TOG CYT-10-169  
 3\_RESYNC\_TOG\_MSTR CYT-10-169  
 3\_SERDES\_ALIGNED CYT-10-169  
 3\_SERDES\_ERROR\_CODE CYT-10-173  
 3\_SERDES\_STATE CYT-10-169  
 3\_STRM\_BAD\_CRC CYT-10-151  
 3\_STRM\_BAD\_CRC\_CM CYT-10-157  
 3\_STRM\_BAD\_CRC\_M CYT-10-154  
 3\_STRM\_NOT\_ENOUGH\_DAT CYT-10-151  
 3\_STRM\_NOT\_ENOUGH\_DAT\_CM CYT-10-157  
 3\_STRM\_NOT\_ENOUGH\_DAT\_M CYT-10-154  
 3\_STRM\_OVERFLOW CYT-10-151  
 3\_STRM\_OVERFLOW\_CM CYT-10-157  
 3\_STRM\_OVERFLOW\_M CYT-10-154  
 3\_STRM\_PKT\_DROP CYT-10-151  
 3\_STRM\_PKT\_DROP\_CM CYT-10-157  
 3\_STRM\_PKT\_DROP\_M CYT-10-154  
 3\_STRM\_TOO\_CMUCH\_DAT\_CM CYT-10-157  
 3\_STRM\_TOO\_MUCH\_DAT CYT-10-151  
 3\_STRM\_TOO\_MUCH\_DAT\_M CYT-10-154  
 3\_TRIG\_ACK\_RCVD CYT-10-150  
 3\_TRIG\_ACK\_RCVD\_CM CYT-10-156  
 3\_TRIG\_ACK\_RCVD\_M CYT-10-153  
 3\_TRIG\_NOMATCH CYT-10-151  
 3\_TRIG\_NOMATCH\_CM CYT-10-157  
 3\_TRIG\_NOMATCH\_M CYT-10-154  
 3\_TRIG\_RCVD CYT-10-150  
 3\_TRIG\_RCVD\_CM CYT-10-156  
 3\_TRIG\_RCVD\_M CYT-10-153  
 3\_UNDER\_CURRENT CYT-10-150  
 3\_UNDER\_CURRENT\_CM CYT-10-156  
 3\_UNDER\_CURRENT\_M CYT-10-153

**A**

ACTIVE CYT-10-187  
 ADDR\_ENCA\_FILTER CYT-6-21  
 ADDR\_ENCB\_FILTER CYT-6-23  
 ADDR\_TRIG\_FILTER CYT-6-19  
 AE\_CON CYT-2-8  
 AE\_FIFO\_OVERFLOW CYT-2-11  
 AE\_RUN\_LEVEL CYT-2-9  
 AE\_STATE CYT-2-11  
 AE\_STATUS CYT-2-10  
 AE\_STREAM\_SEL CYT-2-12  
 Aon Jumpers CYT-13-10  
 Aon-CXP Switches CYT-13-8  
 ATS\_CONDITION CYT-4-18  
 ATS\_CONTROL CYT-4-11  
 ATS\_COUNT CYT-4-17  
 ATS\_CT0\_DEFAULT\_STATE CYT-4-12  
 ATS\_CT1\_DEFAULT\_STATE CYT-4-12  
 ATS\_CT2\_DEFAULT\_STATE CYT-4-12  
 ATS\_CT3\_DEFAULT\_STATE CYT-4-12  
 ATS\_END\_OF\_SEQUENCE CYT-4-18  
 ATS\_IDX\_ACCESS CYT-4-15  
 ATS\_IDX\_JUMP CYT-4-12  
 ATS\_NEXT CYT-4-17  
 ATS\_RESOLUTION CYT-4-17  
 ATS\_RUN\_LEVEL CYT-4-12  
 ATS\_STATE\_CT0 CYT-4-17  
 ATS\_STATE\_CT1 CYT-4-17  
 ATS\_STATE\_CT2 CYT-4-17  
 ATS\_STATE\_CT3 CYT-4-17  
 ATS\_TABLE\_CONTROL CYT-4-14  
 ATS\_TABLE\_ENTRY CYT-4-16  
 ATS\_TRIG\_SEL CYT-4-13

**B**

BFML CYT-1-11  
 BFML documentation CYT-1-11  
 BitBox Output Signal Routing CYT-5-9  
 BITFIELDNAME CYT-P-3  
 BM\_QUADS\_CACHED CYT-3-27  
 BM\_RUN\_LEVEL CYT-3-8  
 BM\_STATE CYT-3-27  
 BOARD\_CONFIG CYT-3-11  
 BOX\_OUT\_DYN\_SEL\_0 CYT-6-26  
 BOX\_OUT\_DYN\_SEL\_1 CYT-6-27  
 BOX\_OUT\_DYN\_SEL\_10 CYT-6-31  
 BOX\_OUT\_DYN\_SEL\_11 CYT-6-31  
 BOX\_OUT\_DYN\_SEL\_2 CYT-6-27

BOX\_OUT\_DYN\_SEL\_3 CYT-6-27  
 BOX\_OUT\_DYN\_SEL\_4 CYT-6-29  
 BOX\_OUT\_DYN\_SEL\_5 CYT-6-29  
 BOX\_OUT\_DYN\_SEL\_6 CYT-6-29  
 BOX\_OUT\_DYN\_SEL\_7 CYT-6-29  
 BOX\_OUT\_DYN\_SEL\_8 CYT-6-31  
 BOX\_OUT\_DYN\_SEL\_9 CYT-6-31  
 BOX\_OUT\_DYN\_SEL\_SET\_A CYT-6-25  
 BOX\_OUT\_DYN\_SEL\_SET\_B CYT-6-28  
 BOX\_OUT\_DYN\_SEL\_SET\_C CYT-6-30  
 BOX\_OUT\_MODE\_DIFF\_0 CYT-6-38  
 BOX\_OUT\_MODE\_DIFF\_1 CYT-6-38  
 BOX\_OUT\_MODE\_DIFF\_10 CYT-6-41  
 BOX\_OUT\_MODE\_DIFF\_11 CYT-6-41  
 BOX\_OUT\_MODE\_DIFF\_2 CYT-6-38  
 BOX\_OUT\_MODE\_DIFF\_3 CYT-6-39  
 BOX\_OUT\_MODE\_DIFF\_4 CYT-6-39  
 BOX\_OUT\_MODE\_DIFF\_5 CYT-6-39  
 BOX\_OUT\_MODE\_DIFF\_6 CYT-6-40  
 BOX\_OUT\_MODE\_DIFF\_7 CYT-6-40  
 BOX\_OUT\_MODE\_DIFF\_8 CYT-6-40  
 BOX\_OUT\_MODE\_DIFF\_9 CYT-6-41  
 BOX\_OUT\_MODE\_OC\_0 CYT-6-45  
 BOX\_OUT\_MODE\_OC\_1 CYT-6-46  
 BOX\_OUT\_MODE\_OC\_2 CYT-6-46  
 BOX\_OUT\_MODE\_OC\_3 CYT-6-46  
 BOX\_OUT\_MODE\_OPTO\_0 CYT-6-43  
 BOX\_OUT\_MODE\_OPTO\_1 CYT-6-43  
 BOX\_OUT\_MODE\_OPTO\_2 CYT-6-43  
 BOX\_OUT\_MODE\_OPTO\_3 CYT-6-44  
 BOX\_OUT\_MODE\_OPTO\_4 CYT-6-44  
 BOX\_OUT\_MODE\_OPTO\_5 CYT-6-44  
 BOX\_OUT\_MODE\_OPTO\_6 CYT-6-45  
 BOX\_OUT\_MODE\_OPTO\_7 CYT-6-45  
 BOX\_OUT\_MODE\_SET\_A CYT-6-32  
 BOX\_OUT\_MODE\_SET\_B CYT-6-37  
 BOX\_OUT\_MODE\_SET\_C CYT-6-42  
 BOX\_OUT\_MODE\_TTL\_0 CYT-6-33  
 BOX\_OUT\_MODE\_TTL\_1 CYT-6-33  
 BOX\_OUT\_MODE\_TTL\_10 CYT-6-36  
 BOX\_OUT\_MODE\_TTL\_11 CYT-6-36  
 BOX\_OUT\_MODE\_TTL\_2 CYT-6-33  
 BOX\_OUT\_MODE\_TTL\_3 CYT-6-34  
 BOX\_OUT\_MODE\_TTL\_4 CYT-6-34  
 BOX\_OUT\_MODE\_TTL\_5 CYT-6-34  
 BOX\_OUT\_MODE\_TTL\_6 CYT-6-35  
 BOX\_OUT\_MODE\_TTL\_7 CYT-6-35  
 BOX\_OUT\_MODE\_TTL\_8 CYT-6-35  
 BOX\_OUT\_MODE\_TTL\_9 CYT-6-36  
 BUF\_MGR\_CON CYT-3-7

BUF\_MGR\_STATUS CYT-3-26  
 BUF\_MGR\_TIMEOUT CYT-3-9  
 Button CYT-13-14

## C

CoaXPress CYT-1-1  
 CoaXPress Power CYT-1-8  
 CON104 CYT-10-2  
 CON105 CYT-10-6  
 CON106 CYT-10-8  
 CON107 CYT-10-10  
 CON108 CYT-10-12  
 CON109 CYT-10-14  
 CON110 CYT-10-16  
 CON111 CYT-10-18  
 CON115 CYT-10-20  
 CON116 CYT-10-22  
 CON117 CYT-10-26  
 CON118 CYT-10-29  
 CON120 CYT-10-32  
 CON121 CYT-10-34  
 CON122 CYT-10-36  
 CON123 CYT-10-38  
 CON127 CYT-10-40  
 CON128 CYT-10-42  
 CON131 CYT-10-44  
 CON136 CYT-10-46  
 CON137 CYT-10-49  
 CON138 CYT-10-51  
 CON139 CYT-10-53  
 CON140 CYT-10-55  
 CON141 CYT-10-57  
 CON142 CYT-10-59  
 CON143 CYT-10-61  
 CON147 CYT-10-63  
 CON148 CYT-10-65  
 CON149 CYT-10-68  
 CON150 CYT-10-71  
 CON152 CYT-10-74  
 CON153 CYT-10-76  
 CON154 CYT-10-78  
 CON155 CYT-10-80  
 CON159 CYT-10-82  
 CON160 CYT-10-84  
 CON163 CYT-10-86  
 CON168 CYT-10-88  
 CON169 CYT-10-91  
 CON170 CYT-10-93  
 CON171 CYT-10-95

CON172 CYT-10-97  
 CON173 CYT-10-99  
 CON174 CYT-10-101  
 CON175 CYT-10-103  
 CON179 CYT-10-105  
 CON180 CYT-10-107  
 CON181 CYT-10-110  
 CON182 CYT-10-113  
 CON184 CYT-10-116  
 CON185 CYT-10-118  
 CON186 CYT-10-120  
 CON187 CYT-10-122  
 CON191 CYT-10-124  
 CON192 CYT-10-126  
 CON195 CYT-10-128  
 CON200 CYT-10-130  
 CON201 CYT-10-133  
 CON202 CYT-10-135  
 CON203 CYT-10-137  
 CON204 CYT-10-139  
 CON205 CYT-10-141  
 CON206 CYT-10-143  
 CON207 CYT-10-145  
 CON211 CYT-10-147  
 CON212 CYT-10-149  
 CON213 CYT-10-152  
 CON214 CYT-10-155  
 CON216 CYT-10-158  
 CON217 CYT-10-160  
 CON218 CYT-10-162, CYT-10-164  
 CON219 CYT-10-166  
 CON223 CYT-10-168  
 CON224 CYT-10-170  
 CON227 CYT-10-172  
 CON356 CYT-10-174  
 CON357 CYT-10-176  
 CON358 CYT-10-178  
 CON360 CYT-10-180  
 CON361 CYT-10-182  
 CON363 CYT-10-184  
 CON372 CYT-10-186  
 CON428 CYT-10-188  
 CON485 CYT-3-3  
 CON486 CYT-3-5  
 CON489 CYT-2-42  
 CON490 CYT-2-45  
 CON548 CYT-2-47  
 CON549 CYT-2-50  
 CON60 CYT-6-2  
 CON61 CYT-6-4  
 CON62 CYT-6-6  
 CON63 CYT-6-10  
 CON64 CYT-6-15  
 CON65 CYT-9-2  
 CON66 CYT-9-4  
 CON67 CYT-9-8  
 CON68 CYT-9-11  
 CON69 CYT-9-13  
 CPL\_ERROR CYT-3-28  
 CPL\_STATUS CYT-3-27  
 CPLD\_MODE CYT-3-12  
 CPLD\_STRAP CYT-3-12  
 CURR\_FETCH\_SIZE CYT-3-8  
 CXP\_OUT\_TABLE\_EN CYT-10-185  
 CXP\_OUT\_TABLE\_EVEN CYT-10-185  
 CXP\_OUT\_TABLE\_LPC CYT-10-185  
 CXP\_OUT\_TABLE\_ODD CYT-10-185  
 Cyton Camera Configuration Files CYT-1-11  
 Cyton Jumpers CYT-13-9  
 Cyton-CXP Switches CYT-13-6

**D**

DISABLE\_PKT\_FLUSH\_TIMER CYT-3-30  
 DISABLE\_PKT\_GEN CYT-3-30  
 DISABLE\_PKT\_TAG\_CHK CYT-10-187  
 DISABLE\_TIMEOUT CYT-3-10  
 DIV\_RESET\_DISABLE CYT-6-8  
 DST\_ADDR\_ERROR\_LSB CYT-3-27

**E**

EN\_ENCA CYT-6-8  
 EN\_ENCB CYT-6-8  
 EN\_TRIG CYT-6-8  
 ENC\_DIV\_AUTO\_RESET\_DISABLE CYT-9-12  
 ENC\_DIV\_FCLK\_SEL CYT-9-10  
 ENC\_DIV\_M CYT-9-3  
 ENC\_DIV\_N CYT-9-3  
 ENC\_DIV\_OPEN\_LOOP CYT-9-10  
 ENC\_DIV\_RESET CYT-9-12  
 ENCA\_POL CYT-6-17  
 ENCB\_FILTER CYT-6-24  
 ENCB\_POL CYT-6-17  
 Encoder Divider CYT-7-1  
 ENINT\_ALL CYT-2-46  
 ENINT\_CXP CYT-6-3

**F**

FIRST\_QUAD\_PTR\_HI CYT-3-6  
 FIRST\_QUAD\_PTR\_LO CYT-3-4  
 FORCE\_RESYNC CYT-10-189  
 FPGA\_ID CYT-10-177  
 FW\_BUILD\_DAY CYT-10-175  
 FW\_BUILD\_HOUR CYT-10-177  
 FW\_BUILD\_MIN CYT-10-177  
 FW\_BUILD\_MONTH CYT-10-175  
 FW\_BUILD\_YEAR CYT-10-175  
 FW\_CMPTBL CYT-10-177

**I**

I/O Connector Pinout the Aon-CXP P6 CYT-13-19  
 I/O Connector Pinout the Cyton-CXP P4 CYT-13-17  
 IH\_ERR CYT-10-189  
 INT\_AE\_LOSS\_OF\_SYNC CYT-2-44  
 INT\_AE\_LOSS\_OF\_SYNC\_M CYT-2-49  
 INT\_AE\_LOSS\_OF\_SYNC\_WP CYT-2-52  
 INT\_ANY CYT-2-46  
 INT\_BM\_ERROR CYT-2-43  
 INT\_BM\_ERROR\_M CYT-2-48  
 INT\_BM\_ERROR\_WP CYT-2-51  
 INT\_CXP CYT-6-3  
 INT\_ENC\_A CYT-2-43  
 INT\_ENC\_A\_M CYT-2-48  
 INT\_ENC\_A\_WP CYT-2-51  
 INT\_ENC\_B CYT-2-43  
 INT\_ENC\_B\_M CYT-2-48  
 INT\_ENC\_B\_WP CYT-2-51  
 INT\_PCIE\_PKT\_DROPPED CYT-2-44  
 INT\_PCIE\_PKT\_DROPPED\_M CYT-2-49  
 INT\_PCIE\_PKT\_DROPPED\_WP CYT-2-52  
 INT\_TRIG CYT-2-43  
 INT\_TRIG\_M CYT-2-48  
 INT\_TRIG\_WP CYT-2-51  
 INT\_V\_ACQUIRED CYT-2-43  
 INT\_V\_ACQUIRED\_M CYT-2-48  
 INT\_V\_ACQUIRED\_WP CYT-2-51  
 INT\_V\_START\_M CYT-2-48  
 INT\_V\_START\_WP CYT-2-51  
 INT\_Y\_ACQUIRED CYT-2-43  
 INT\_Y\_ACQUIRED\_M CYT-2-48  
 INT\_Y\_ACQUIRED\_WP CYT-2-51  
 INT\_Y\_START\_M CYT-2-48  
 INT\_Y\_START\_WP CYT-2-51

INT\_Z\_ACQUIRED CYT-2-43  
 INT\_Z\_ACQUIRED\_LEGACY CYT-2-44  
 INT\_Z\_ACQUIRED\_LEGACY\_M CYT-2-49  
 INT\_Z\_ACQUIRED\_LEGACY\_WP CYT-2-52  
 INT\_Z\_ACQUIRED\_M CYT-2-48  
 INT\_Z\_ACQUIRED\_WP CYT-2-51  
 INT\_Z\_START\_M CYT-2-48  
 INT\_Z\_START\_WP CYT-2-51  
 IS\_LINE\_SCAN CYT-10-189

**L**

LED\_GREEN CYT-6-18  
 LED\_ORANGE CYT-6-18  
 LED\_RED CYT-6-17  
 LINK\_0\_ORDER CYT-10-183  
 LINK\_1\_ORDER CYT-10-183  
 LINK\_2\_ORDER CYT-10-183  
 LINK\_3\_ORDER CYT-10-183

**M**

MAX\_FETCH\_SIZE CYT-3-8  
 MAX\_PAYLOAD\_PCIE CYT-3-30  
 MAX\_PAYLOAD\_USER CYT-3-30

**N**

NEW\_FRAME\_RESYNC CYT-3-20  
 NEXT\_ADDR\_ERROR\_LSB CYT-3-27  
 NEXT\_LINK CYT-10-187  
 NO\_QUAD\_AVAIL CYT-3-20  
 NUM\_LINKS\_NEEDED CYT-10-181  
 NUM\_PACKETS\_DROP CYT-3-14  
 NUM\_PACKETS\_SENT CYT-3-14  
 NUM\_QTABS\_LOADED CYT-3-25  
 NUM\_QTABS\_USED CYT-3-18  
 NUM\_QUADS\_LOADED CYT-3-23  
 NUM\_QUADS\_USED CYT-3-16

**P**

PACK\_PIXELS CYT-10-189  
 PACKETS\_SENT\_STATUS CYT-3-13  
 PFG\_RESET CYT-10-179  
 PKT\_CON CYT-3-29  
 PKT\_FLUSH\_ENABLE CYT-3-21  
 PKT\_GNT\_CNT CYT-10-187  
 PKT\_STAT CYT-3-19  
 PKT\_STATE CYT-3-20  
 PoCXP power, Aon CYT-13-10

PoCXP power, Cyton CYT-13-9

## Q

QENC\_AQ\_DIR CYT-9-5  
 QENC\_COUNT CYT-9-14  
 QENC\_DECODE CYT-9-5  
 QENC\_DIR CYT-9-14  
 QENC\_DUAL\_PHASE CYT-9-6  
 QENC\_INTRVL\_IN CYT-9-14  
 QENC\_INTRVL\_LL CYT-9-5  
 QENC\_INTRVL\_MODE CYT-9-5  
 QENC\_INTRVL\_UL CYT-9-9  
 QENC\_NEW\_LINES CYT-9-15  
 QENC\_NO\_REAQ CYT-9-5  
 QENC\_PHASEA CYT-9-12, CYT-9-14  
 QENC\_PHASEB CYT-9-14, CYT-9-15  
 QENC\_REAQ\_MODE CYT-9-9  
 QENC\_RESET CYT-9-7  
 QENC\_RESET\_REAQ CYT-9-9  
 QTABS\_LOADED\_STATUS CYT-3-24  
 QTABS\_USED\_STATUS CYT-3-17  
 QUAD\_COMPLETE\_TIMEOUT CYT-3-10  
 QUAD\_DROPPED CYT-3-20  
 QUAD\_FIFO\_OVERFLOW CYT-3-28  
 QUAD\_NUM\_MISMATCH CYT-3-28  
 QUAD\_TIMEOUT\_DETECTED CYT-3-28  
 QUADS\_LOADED\_STATUS CYT-3-22  
 QUADS\_USED\_STATUS CYT-3-15

## R

R/W CYT-P-4  
 RD\_BOX\_IN\_DIF CYT-6-3  
 RD\_BOX\_IN\_OPTO CYT-6-5  
 RD\_BOX\_IN\_TTL CYT-6-3, CYT-6-22  
 RD\_BUTTON CYT-6-8  
 RD\_CXP\_TRIG\_IN CYT-6-8  
 RD\_CXP\_TRIG\_OUT CYT-6-5  
 RD\_ENCA\_DIF CYT-6-7  
 RD\_ENCA\_SELECTED CYT-6-9  
 RD\_ENCA\_SW CYT-6-7  
 RD\_ENCA\_TTL CYT-6-7  
 RD\_ENCA\_VFG0 CYT-6-7  
 RD\_ENCB\_DIF CYT-6-8  
 RD\_ENCB\_SELECTED CYT-6-8  
 RD\_ENCB\_SW CYT-6-8  
 RD\_ENCB\_TTL CYT-6-7  
 RD\_ENCB\_VFG0 CYT-6-8  
 RD\_ENCO\_SELECTED CYT-9-12

RD\_ON\_EMPTY CYT-3-20  
 RD\_SCAN\_STEP CYT-6-7  
 RD\_SW\_TRIG CYT-6-7  
 RD\_TRIG\_DIF CYT-6-7  
 RD\_TRIG\_SELECTED CYT-6-9  
 RD\_TRIG\_TTL CYT-6-7  
 RD\_TRIG\_VFG0 CYT-6-7  
 RO CYT-P-4

## S

SCAN\_STEP CYT-9-3  
 SCAN\_STEP\_TRIG CYT-9-6  
 SEL\_CC1 CYT-6-13  
 SEL\_CC2 CYT-6-13  
 SEL\_CC3 CYT-6-16  
 SEL\_CC4 CYT-6-16  
 SEL\_ENCA CYT-6-11  
 SEL\_ENCB CYT-6-12  
 SEL\_ENCDIV CYT-9-3  
 SEL\_ENCDIV\_INPUT CYT-9-3  
 SEL\_ENCO CYT-9-3  
 SEL\_LED CYT-6-14  
 SEL\_TRIG CYT-6-11  
 SF\_CON CYT-2-55  
 SF\_DIM CYT-2-53  
 SF\_HEIGHT CYT-2-54  
 SF\_INC\_X CYT-2-57  
 SF\_INC\_Y CYT-2-57  
 SF\_INC\_Z CYT-2-57  
 SF\_INIT\_BYTE CYT-2-56  
 SF\_LINE\_SCAN CYT-2-56  
 SF\_MODE CYT-2-56  
 SF\_RUN\_LEVEL CYT-2-56  
 SF\_STATE CYT-2-56  
 SF\_WIDTH CYT-2-54  
 SF\_X\_GAP CYT-2-56  
 SF\_Y\_GAP CYT-2-57  
 SF\_Z\_GAP CYT-2-57  
 SIZE\_ERROR\_LSB CYT-3-27  
 SIZE\_ERROR\_MSB CYT-3-27  
 SOF\_ERR CYT-10-189  
 SOL\_ERR CYT-10-189  
 SP\_ARM Manual-11-10  
 SP\_BUSY Manual-11-11  
 SP\_CON Manual-11-9  
 SP\_COUNT Manual-11-13  
 SP\_COUNT\_MODE Manual-11-10  
 SP\_COUNT\_UPDATED Manual-11-10  
 SP\_EVENTS Manual-11-6

SP\_LIMIT Manual-11-14  
 SP\_RST Manual-11-10  
 SP\_START\_EVENT Manual-11-7  
 SP\_START\_FUNC Manual-11-8  
 SP\_STAT Manual-11-12  
 SP\_STOP\_EVENT Manual-11-7  
 SP\_STOP\_EVENT\_LIMIT Manual-11-15  
 SP\_STOP\_FUNC Manual-11-7  
 SP\_SYNC\_EVENT Manual-11-8  
 SP\_SYNC\_FUNC Manual-11-8  
 SP\_TARGET\_EVENT Manual-11-7  
 SP\_TARGET\_FUNC Manual-11-7  
 SP\_WRAP\_COUNT Manual-11-10  
 Specifications CYT-12-1, CYT-12-2  
 SRC\_TAG\_ERR CYT-10-189  
 STREAM\_SEL CYT-2-13  
 STRM\_DECODE\_STATE CYT-10-189  
 SW CYT-3-12  
 SW\_ENCA CYT-6-3  
 SW\_ENCB CYT-6-3  
 SW\_TRIG CYT-6-3  
 System Probe Manual-11-1

**T**

The Auxiliary Power Connector CYT-13-15  
 The Cyton LEDs CYT-13-11, CYT-13-13  
 The Stream Sync DMA Engine CYT-1-8  
 The Timing Sequencer Signal Generator CYT-1-7  
 Timing Sequencer CYT-4-1  
 TRIG\_FILTER CYT-6-20, CYT-6-22, CYT-6-24  
 TRIGPOL CYT-6-17  
 TS CYT-4-1  
 TS\_CONDITION CYT-4-10  
 TS\_CONTROL CYT-4-3  
 TS\_COUNT CYT-4-9  
 TS\_CT0\_DEFAULT\_STATE CYT-4-4  
 TS\_CT1\_DEFAULT\_STATE CYT-4-4  
 TS\_CT2\_DEFAULT\_STATE CYT-4-4  
 TS\_CT3\_DEFAULT\_STATE CYT-4-4  
 TS\_END\_OF\_SEQUENCE CYT-4-10  
 TS\_IDX\_ACCESS CYT-4-7  
 TS\_IDX\_JUMP CYT-4-4  
 TS\_NEXT CYT-4-9  
 TS\_RESOLUTION CYT-4-9  
 TS\_RUN\_LEVEL CYT-4-4, CYT-4-5  
 TS\_STATE\_CT0 CYT-4-9  
 TS\_STATE\_CT1 CYT-4-9  
 TS\_STATE\_CT2 CYT-4-9

TS\_STATE\_CT3 CYT-4-9  
 TS\_TABLE\_CONTROL CYT-4-6  
 TS\_TABLE\_ENTRY CYT-4-8  
 TS\_TERMINATE CYT-4-10, CYT-4-18  
 TS\_TRIG\_SEL CYT-4-5

**U**

UNEXP\_PKT\_TAG\_CNT CYT-10-187  
 USE\_SYNTHETIC\_FRAME CYT-2-13

**V**

V\_ACQ\_COUNT CYT-2-35  
 V\_ACQ\_COUNT\_CLR\_MODE CYT-2-35, CYT-2-39  
 V\_ACQUIRED CYT-2-34  
 V\_SIZE CYT-2-15  
 V\_WIN\_DIM CYT-2-14  
 VFG CYT-1-2  
 VIDEO\_DROPPED CYT-3-20

**W**

WO CYT-P-4  
 WR\_ON\_FULL CYT-3-20

**X**

X\_ACQ\_COUNT CYT-2-41  
 X\_ACQ\_COUNT\_CLR\_MODE CYT-2-41  
 X\_ACQUIRED CYT-2-40  
 X\_OFFS CYT-2-33  
 X\_SIZE CYT-2-33  
 X\_WIN\_DIM CYT-2-32

**Y**

Y\_ACQ\_COUNT CYT-2-39  
 Y\_ACQUIRED CYT-2-38  
 Y\_CLOSE CYT-2-25  
 Y\_CLOSE\_TRIG\_FUNC CYT-2-25  
 Y\_CLOSE\_TRIG\_SEL CYT-2-25  
 Y\_OFFS CYT-2-29  
 Y\_OPEN\_TRIG\_FUNC CYT-2-26  
 Y\_OPEN\_TRIG\_SEL CYT-2-26  
 Y\_SIZE CYT-2-29  
 Y\_SIZE\_MSB CYT-2-31  
 Y\_SYNC CYT-2-27  
 Y\_WIN\_CON CYT-2-24  
 Y\_WIN\_DIM CYT-2-28



Y\_WIN\_DIM\_EX CYT-2-30

**Z**

Z\_ACQ\_COUNT CYT-2-37

Z\_ACQ\_COUNT\_CLR\_MODE CYT-2-37

Z\_ACQUIRED CYT-2-36

Z\_CLOSE CYT-2-17

Z\_CLOSE\_TRIG\_FUNC CYT-2-17

Z\_CLOSE\_TRIG\_SEL CYT-2-17

Z\_OFFS CYT-2-21

Z\_OPEN CYT-2-18, CYT-2-26

Z\_OPEN\_TRIG\_FUNC CYT-2-18

Z\_OPEN\_TRIG\_SEL CYT-2-18

Z\_SIZE CYT-2-21

Z\_SIZE\_MSB CYT-2-23

Z\_SYNC CYT-2-19

Z\_WIN\_CON CYT-2-16

Z\_WIN\_DIM CYT-2-20

Z\_WIN\_DIM\_EX CYT-2-22









