Japan Industrial Imaging Association Standard
日本インダストリアルイメージング協会規格

JIIA CXP-001-2015

CoaXPress Standard
CoaXPress 規格書

Version 1.1.1
Revised on Dec 29, 2015
2015年12月29日改正

The Standardization Committee — CoaXPress Working Group
Japan Industrial Imaging Association
標準化委員会 — CoaXPress 分科会
一般社団法人 日本インダストリアルイメージング協会
This document is provided “as it is” with this current version.

None of JIIA, its members including Liaison members, the subsidiary companies of its members, or the affiliates of its members are liable, whether express or implied, for any kinds of warranties of merchantability, fitness for a particular purpose, and non-infringement of third party property rights.

JIIA, its members including Liaison members, the subsidiary companies of its members, or the affiliates of its members are exempted from taking responsibility and held harmless under the applicable law, whether express or implied, for any kinds of damages or losses arising from a course of usage of this document, including but not limited to passive damages, derivative damages, and incidental damages. This is applied even when any of JIIA, its members including Liaison members, the subsidiary companies of its members, or the affiliates of its members was informed of the possibilities of damages in advance.

JIIA, its members including Liaison members, the subsidiary companies of its members, or the affiliates of its members are exempted from taking responsibility and held harmless under the applicable law, whether express or implied, for any kinds of damages or losses arising from a course of usage of this document, including but not limited to passive damages, derivative damages, and incidental damages. This is applied even when any of JIIA, its members including Liaison members, the subsidiary companies of its members, or the affiliates of its members was informed of the possibilities of damages in advance.

The Japanese version of this document is a translation from the original English. In the event of doubt about the intended meaning, the English version is definitive.

CoaXPress is a registered trademark in Japan, the USA, the EU and other countries and the property of the trademark belongs to Japan Industrial Imaging Association (JIIA). You need a trademark license to use the logo on any product or in commercial documents. If you want to use the trademark or have any questions about using trademarks, please contact a JIIA representative (sec@jiia.org).

© 2015 Japan Industrial Imaging Association
# TABLE OF CONTENTS

1 Scope ................................................................................................................................. 1
2 References ............................................................................................................................ 1
  2.1 Normative References ............................................................................................... 1
  2.2 General References ..................................................................................................... 1
3 References and Definitions of Terms .................................................................................. 3
  3.1 Definition of Terms ................................................................................................. 3
  3.2 Abbreviations ........................................................................................................... 4
  3.3 Conventions ............................................................................................................... 4
4 Introduction to CoaXPress ............................................................................................... 5
  4.1 Overview .................................................................................................................... 5
  4.2 Use of Coax Cable .................................................................................................... 6
  4.3 CoaXPress Speeds .................................................................................................. 7
  4.4 Connection Aggregation ......................................................................................... 7
  4.5 Maximum Cable Length .......................................................................................... 7
  4.6 Data Communication ............................................................................................... 8
  4.7 Trigger Accuracy ...................................................................................................... 8
  4.8 Power over Cable ................................................................................................. 8
  4.9 Data Integrity ........................................................................................................... 8
  4.10 Plug and Play ....................................................................................................... 9
    4.10.1 General ........................................................................................................... 9
    4.10.2 GenICam ....................................................................................................... 9
  4.11 Labeling and Use of the CoaXPress Logos ............................................................... 9
    4.11.1 Official CoaXPress Logo ............................................................................. 9
    4.11.2 CoaXPress Logo Varieties ......................................................................... 10
    4.11.3 Product Labeling ........................................................................................ 10
    4.11.4 Product Feature Bar .................................................................................... 11
5 Cable and Connector Specification .................................................................................... 12
  5.1 Introduction ............................................................................................................... 12
  5.2 Connectors ................................................................................................................ 12
    5.2.1 Single Connectors .......................................................................................... 12
    5.2.2 Multi-Connectors .......................................................................................... 13
    5.2.3 Couplers ......................................................................................................... 14
    5.2.4 Contact Material .......................................................................................... 14
  5.3 Multiple Cables for Connection Aggregation ............................................................ 14
B.4.4 Low Speed Cable Receiver .............................................................. 112
B.5 Guidance Notes .................................................................................. 112

Annex C: Descriptive Clause .................................................................... 113

C.1 Background of Standardization ............................................................ 113
C.1.1 Background of the Original CoaXPress Technology ....................... 113
C.1.2 Handover from the CoaXPress Consortium to JIIA ......................... 113
C.2 Version Changes .................................................................................. 114
C.2.1 v1.0 to v1.1 ..................................................................................... 114
C.2.2 v1.1 to v1.1.1 ................................................................................. 115
C.3 Members Involved ............................................................................... 116
C.3.1 The Main Authors of the Original CoaXPress Specification ............ 116
C.3.2 The Japanese Translators of the Original CoaXPress Specification .. 116
C.3.3 JIIA CoaXPress WG Members for v1.0 ........................................... 117
C.3.4 CoaXPress Liaison Members for v1.0 ............................................. 118
C.3.5 Contributors for v1.1 ....................................................................... 118
C.3.6 Contributors for v1.1.1 .................................................................. 119
TABLE OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CoaXPress physical topology</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Signaling connections and data flow</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>Official CoaXPress Logo</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>CoaXPress logos</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>Feature bar</td>
<td>11</td>
</tr>
<tr>
<td>6</td>
<td>Device / Host Multi-Connector</td>
<td>13</td>
</tr>
<tr>
<td>7</td>
<td>Multi-Connector Locking Slot</td>
<td>14</td>
</tr>
<tr>
<td>8</td>
<td>Electrical block diagram</td>
<td>17</td>
</tr>
<tr>
<td>9</td>
<td>Definition of the jitter in the transmit EYE at Tp2</td>
<td>20</td>
</tr>
<tr>
<td>10</td>
<td>Definition of the jitter in the low speed transmit EYE at Tp3</td>
<td>21</td>
</tr>
<tr>
<td>11</td>
<td>PoCXP block diagram</td>
<td>25</td>
</tr>
<tr>
<td>12</td>
<td>Simplified example state diagram</td>
<td>30</td>
</tr>
<tr>
<td>13</td>
<td>Link Overview</td>
<td>31</td>
</tr>
<tr>
<td>14</td>
<td>Link Overview with High Speed Upconnection</td>
<td>32</td>
</tr>
<tr>
<td>15</td>
<td>Bit Order</td>
<td>33</td>
</tr>
<tr>
<td>16</td>
<td>CRC Calculation</td>
<td>35</td>
</tr>
<tr>
<td>17</td>
<td>High speed connection packet transmission example</td>
<td>37</td>
</tr>
<tr>
<td>18</td>
<td>Low speed connection packet transmission example</td>
<td>38</td>
</tr>
<tr>
<td>19</td>
<td>Low speed connection trigger insertion example</td>
<td>38</td>
</tr>
<tr>
<td>20</td>
<td>Low speed trigger example</td>
<td>41</td>
</tr>
<tr>
<td>21</td>
<td>Packet formation in Device – image data example</td>
<td>45</td>
</tr>
<tr>
<td>22</td>
<td>Packet processing in Device</td>
<td>47</td>
</tr>
<tr>
<td>23</td>
<td>Control transaction – standard</td>
<td>49</td>
</tr>
<tr>
<td>24</td>
<td>Control transaction – wait acknowledgment</td>
<td>49</td>
</tr>
<tr>
<td>25</td>
<td>Connection test methodology</td>
<td>53</td>
</tr>
<tr>
<td>26</td>
<td>Image stream structure</td>
<td>57</td>
</tr>
<tr>
<td>27</td>
<td>Packing of 8 bit pixels or components</td>
<td>65</td>
</tr>
<tr>
<td>28</td>
<td>Packing of 10 bit pixels or components</td>
<td>65</td>
</tr>
<tr>
<td>29</td>
<td>Packing of 12 bit pixels or components</td>
<td>66</td>
</tr>
<tr>
<td>30</td>
<td>Packing of 14 bit pixels or components</td>
<td>66</td>
</tr>
<tr>
<td>31</td>
<td>Packing of 16 bit pixels or components</td>
<td>66</td>
</tr>
<tr>
<td>32</td>
<td>Packing of 15 into 16 bits</td>
<td>67</td>
</tr>
<tr>
<td>33</td>
<td>Scanning example</td>
<td>68</td>
</tr>
<tr>
<td>34</td>
<td>Example 8x8 ROI in a VGA sized sensor</td>
<td>72</td>
</tr>
</tbody>
</table>
Figure 35 — Matching Device and Host bit rates ................................................................. 79
Figure 36 — Examples of Device – Host connection schemes .............................................. 80
Figure 37 — Connection topology example part 1 ................................................................. 81
Figure 38 — Connection topology example part 2 ................................................................. 81
Figure 39 — Setting the bit rate ............................................................................................ 85
Figure 40 — GenICam Terminology ..................................................................................... 96

Annex A: Figure 1 — Deviation from normal attenuation behavior versus frequency .......... 104
Annex A: Figure 2 — CXP-multi-cable rules ........................................................................ 105

Annex B: Figure 1 — Electrical block diagram ................................................................... 106
Annex B: Figure 2 — Definition of the parameters in the transmit EYE at Tp2 .................. 107
Annex B: Figure 3 — Signal transmitted at Tp3 by the Host showing baseline wander ....... 110
Annex B: Figure 4 — Definition of the parameters in the low speed transmit EYE at Tp3 ... 111
TABLE OF TABLES

Table 1 — Supported high speed connection bit rates ......................................................... 7
Table 2 — Labeling of CoaXPress products depending on their allowed bit rate .................... 10
Table 3 — Connector indicator lamp states ............................................................................ 15
Table 4 — Connector indicator lamp timings ......................................................................... 15
Table 5 — Supported high speed connection bit rates ......................................................... 19
Table 6 — Supported low speed connection bit rate ............................................................. 20
Table 7 — Normative return loss frequency ranges for Host .................................................. 22
Table 8 — Normative return loss frequency ranges for Device .............................................. 22
Table 9 — Key to Host block diagram ................................................................................... 25
Table 10 — Key to Device block diagram .............................................................................. 25
Table 11 — K-code usage ...................................................................................................... 33
Table 12 — Packet types ....................................................................................................... 36
Table 13 — Packet transmission priority ................................................................................ 37
Table 14 — Idle word format ................................................................................................... 39
Table 15 — Low speed connection trigger packet format ......................................................... 42
Table 16 — High speed connection trigger packet format ......................................................... 42
Table 17 — I/O acknowledgment packet format ...................................................................... 43
Table 18 — Data packet transmission format .......................................................................... 44
Table 19 — Stream data packet format .................................................................................... 46
Table 20 — Packet order on connections – example ................................................................. 48
Table 21 — Control command payload format ......................................................................... 51
Table 22 — Acknowledgment message format ........................................................................ 52
Table 23 — Connection test packet format ............................................................................. 54
Table 24 — PixelF coding ........................................................................................................ 58
Table 25 — PixelF values ......................................................................................................... 58
Table 26 — Data width definition ............................................................................................. 61
Table 27 — Mono definition .................................................................................................... 61
Table 28 — Planar definition ................................................................................................... 61
Table 29 — Bayer definition ................................................................................................... 62
Table 30 — RGB definition .................................................................................................... 62
Table 31 — RGBA definition .................................................................................................. 62
Table 32 — YUV definition ..................................................................................................... 63
Table 33 — YCbCr601 definition ............................................................................................. 63
Table 34 — YCbCr709 definition ............................................................................................. 64
Table 35 — Vertical tap formats ........................................................................................................68
Table 36 — TapG coding ......................................................................................................................69
Table 37 — TapG values .......................................................................................................................69
Table 38 — Rectangular image header ..............................................................................................71
Table 39 — Rectangular image line marker .......................................................................................73
Table 40 — Arbitrary image header ..................................................................................................75
Table 41 — Arbitrary line marker ......................................................................................................76
Table 42 — Connection state conditions ...........................................................................................77
Table 43 — Control packet size negotiation .......................................................................................83
Table 44 — Stream packet size negotiation .......................................................................................83
Table 45 — Bootstrap registers ..........................................................................................................87
Table 46 — Bit rate codes ..................................................................................................................93
Table 47 — Mandatory use case features and registers ....................................................................97

Annex A: Table 1 — Return loss specification depending on frequency range .................................101
Annex A: Table 2 — Specification of the maximum attenuation for a cable .....................................102
Annex A: Table 3 — Frequency range in which cable-like behavior is to be guaranteed ...............103

Annex B: Table 1 — High speed connection parameters at the transmit (Device) side ..................108
Annex B: Table 2 — High speed connection parameters at the receiver (Host) side ......................109
Annex B: Table 3 — Low speed connection parameters at the transmit (Host) side .......................111
Annex B: Table 4 — Low speed connection parameters at the receive (Device) side .....................112
Foreword

JIIA (Japan Industrial Imaging Association) is the Japanese machine vision standards body.
The initial development of CoaXPress was by the CoaXPress Consortium. The JIIA CoaXPress Technical Committee is responsible for the preparation and maintenance of this standard. Contributions to the standard are also made by the CoaXPress Consortium and the CoaXPress Liaison Group, which is for members of the AIA (formerly Automated Imaging Association) and EMVA (European Machine Vision Association).
1 Scope

The CoaXPress digital interface was developed for high speed image data transmission and intended mainly for Machine Vision applications. The interface is also suitable for other imaging applications and for high speed data transmission in other fields.

CoaXPress uses 75 Ω coaxial cable as a physical medium.

2 References

2.1 Normative References

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

- IEC 61169-8 Ed. 1.0:2007, Radio-frequency connectors — Part 8: Sectional specification — RF coaxial connectors with inner diameter of outer conductor 6.5 mm (0.256 in) with bayonet lock — Characteristic impedance 50 Ohm (type BNC).
- IEC 61169-29 Ed. 1.0:2005, Radio-frequency connectors — Part 29: Sectional specification — Miniature radio frequency coaxial connectors model screw, snap-on, push-pull or quick-lock, slide-in (rack and panel applications ) — Characteristic impedance 50 Ω (type 1,0/2,3) – 50 Ω and 75 Ω applications.
- GenICam™ standard from the European Machine Vision (EMVA) website (http://www.emva.org).
  See the GenICam section in this specification for version requirements.
- Byte oriented DC balanced 8B/10B partitioned block transmission code. US Patent 4486739.

2.2 General References

- Ref 2 Technical specification for Belden 1694A cable from the Belden website.
- Ref 3 ZIP format as defined on the PKWARE website (http://www.pkware.com/documents/casestudies/APPNOTE.TXT).
- Ref 5 Recommendation ITU-R BT.709-5 “Parameter values for the HDTV standard for production and international programme exchange”.

CoaXPress Standard Version 1.1.1 1 JIIA CXP-001-2015
• Ref 6  IIDC2 Digital Camera Control Specification from the Japan Industrial Imaging Association (JIIA) website (http://www.jiia.org).

• Ref 7  Pixel Format Naming Convention (PFNC) from the EMVA website (http://www.emva.org).

• Ref 8  CoaXPress product registration and test procedures from the Japan Industrial Imaging Association (JIIA) website (http://www.jiia.org).
3 References and Definitions of Terms

3.1 Definition of Terms

Device  Can be any type of system that generates and transmits images or high speed data, as long as it is further responding and acting according to the CoaXPress specification. In most cases the Device will be a digital camera. A Device can also be on a frame grabber to allow for daisy chaining or data forwarding.

Host  Can be any type of system that receives, records, processes or displays high speed data, as long as it is further responding and acting in accordance with the CoaXPress specification. In most cases the Host will be a simple interface card, or more powerful frame grabber or image processor.

Connection  A single coaxial cable that connects a Device to a Host, which has the ability to provide a high speed connection in one direction, a low speed connection in the opposite direction, and power.

Link  One or more connections, comprising one Master connection and zero or more extension connections.
3.2 Abbreviations

CXP  CoaXPress
DT   Device Transceiver circuit
Gbps Giga bit per second, $10^9$ bits per second
HT   Host Transceiver circuit
Mbps Mega bit per second, $10^6$ bits per second
OCP  Over Current Protection
PoCXP Power over CoaXPress
ppm  parts per million
PRU  Power Receiving Unit
PTU  Power Transmitting Unit
ROI  Region of Interest

3.3 Conventions

“Shall” means a mandatory requirement.

“Can” means an optional feature.

Comments written in italic text do not form part of the specification, but are intended to help understand the requirements of the specification.
4 Introduction to CoaXPress

4.1 Overview

CoaXPress is an interface to connect Devices (typically cameras) to Hosts (typically frame grabbers). It combines the simplicity of coaxial cable with state of the art high speed serial data technology, allowing up to 6.25 Gbps data rate per cable, plus device control and power in the same cable.

CoaXPress is a point to point scalable interface. The physical medium between the Device and Host is 75Ω coaxial cable.

An interface consists of one master connection and optional extension connections, which together form a link. Each connection is associated with a coax cable. At the Device connections are numbered; 0 for Master, and 1 to \((n-1)\) for \((n-1)\) extension connections as shown in the following figure:

![CoaXPress physical topology](image)

**Figure 1 — CoaXPress physical topology**

Each connection provides the following signaling functions:

- High speed serial (usually Device to Host downconnection), at up to 6.25 Gbps.
- Low speed serial (usually Host to Device upconnection), at \(20.83\) Mbps \(^1\).
- Power (Host to Device), up to 13W.

A dedicated high speed connection from the Host to the Device is allowed for high speed triggers and camera control. This connection does not support power.

\(^1\) Note – the dot after the number 3 means “3 recurring”, i.e. the bitrate is 20.833333… Mbps.
The link protocol defines the transfer of triggers, general purpose I/O, control data and high speed streaming data over a link, as shown in the following figure:

![Diagram of signaling connections and data flow]

Both the upconnection and downconnection use 8B/10B coding.

Comment: 8B/10B is an industry-standard code that maps 8 bit data to 10 bit data to achieve DC balance on the connection, while also allowing clock recovery by ensuring regular transitions.

### 4.2 Use of Coax Cable

There are several reasons for specifying coax:

- Coax is acknowledged to be the best electrical medium for high speed data cables.
- Coax does not suffer from intra-pair skew. Intra-pair skew is an important performance limiter for differential cables at high speeds.
- Optimal bandwidth: a single-ended medium has less than half the loss of a shielded differential pair of the same diameter.
- Very good EMI/EMC performance.
- A large variety of possible cables allowed: thick ones for large distance, thin or flexible for shorter distance.
- It is a low cost solution.
- It is easy to install and terminate connectors in the field.
- Terminating several coaxial cables in a common connector is relatively easy to achieve; terminating several shielded twisted pairs in a common connector is an expensive manual time consuming task.
- There is a legacy cable infrastructure.

In this way using coax cabling, whether a single coax or a set of coax cables is used, optimal performance is always achieved.
4.3 CoaXPress Speeds

CoaXPress can operate at the speeds defined in the following table:

<table>
<thead>
<tr>
<th>Bit Rate (Gbps)</th>
<th>Discovery Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.250</td>
<td>✔</td>
</tr>
<tr>
<td>2.500</td>
<td></td>
</tr>
<tr>
<td>3.125</td>
<td>✔</td>
</tr>
<tr>
<td>5.000</td>
<td></td>
</tr>
<tr>
<td>6.250</td>
<td></td>
</tr>
</tbody>
</table>

Comment: Additional speeds, both lower and higher, may be defined in future revisions of the specification.

A Device (camera) shall support at least one of the defined bit rates in normal operation, plus a discovery rate (see section 10.1.2).

Comment: Usually the bit rate will be fixed, and will be the lowest speed that can be used to suite the Device's performance. Unnecessary use of higher speeds will result in higher power consumption and will also reduce maximum cable length. The number of discovery bit rates is restricted to a small number of values to speed up the discovery process. More than one discovery bit rate is defined because a Device's SERDES may not support 1.25 Gbps, especially as higher speeds are added in future.

A CoaXPress Host (frame grabber) shall support all the defined bit rates between 1.25 Gbps and the highest rate it supports.

CoaXPress products shall be labeled to indicate their maximum speed as described in section 4.11.3.

4.4 Connection Aggregation

CoaXPress Devices can use more than one coax cable in order to increase the maximum speed as described in section 4.1 above. The specification does not limit the number of cables which makes CoaXPress a future proof interface.

Comment: CoaXPress Devices slower than 6.25 Gbps can also be used with more than one cable. In some situations two cables running at 3.125 Gbps may be preferred to one cable at 6.25 Gbps as it will allow for longer cable length and may result in a lower system cost (because lower cost components can be used).

4.5 Maximum Cable Length

The maximum cable length between Device and Host is dependent on the bit rate and the type of coax cable. For this reason CoaXPress certified cables are clearly marked with the maximum speed they support.

Example: At 6.25 Gbps the maximum cable length is around 25 meters for thin or highly flexible cables. At the other extreme, lengths of over 200 meters are possible with thicker cables at 1.25 Gbps.

Comment: The connection test mode gives an indication of the error rate on the connection as a part of a diagnostics test, or for testing if an existing cable in an analog system can be reused for CoaXPress.
4.6 Data Communication
Trigger, control data and streaming data share bandwidth. Different types of data are transferred using a priority scheme. As a result of this priority scheme the bandwidth available for low priority data is automatically limited by the amount of used bandwidth for high priority data.

4.7 Trigger Accuracy
Trigger sequences have the highest priority. Triggers on a high speed connection have an accuracy equal to ten transmitter bit periods, that is, better than ±2 ns at 3.125 Gbps.
Triggers on a low speed connection use an accurate timing correction value in the trigger words that allows for a low fixed trigger latency of 3.4 µs, with an accuracy of ±4 ns.
The optional high speed upconnection provides a lower latency and higher accuracy trigger from the Host to the Device.

4.8 Power over Cable
The available power per cable is 13W at the Device, at a nominal 24V.
It is anticipated that 13W should be enough for most Devices with one CoaXPress coax connector; likewise 26W should be sufficient for Devices with two connectors (up to 12.5 Gbps); 39W for three connectors, etc.
Should more power be needed, or if the Device is not designed to be powered over the cable, a separate power input on the Device can be used.
CoaXPress Hosts are equipped with Device detection and short circuit protection to minimize the risk of damage should a device other than a CoaXPress Device be connected.
When the optional high speed upconnection is supported, it does not provide power.

4.9 Data Integrity
The control protocol (i.e. everything except for data) is designed to be tolerant of single bit errors. The data itself is protected by a CRC32 checksum which signals data errors. This revision of the protocol does not support data resend.

Comment: Data resend is not supported in the current revision because CoaXPress uses a physical medium that is designed to be nominally error free. Adding a resend capability will significantly increase the complexity of the protocol, add costs to the Devices that support it and decrease the real-time performance. CoaXPress can however inform the application whether there are bit errors and then an appropriate course of action taken.
For off-line testing of the complete imaging chain (Device – Cable(s) – Host), CoaXPress has a connection test mode. In this mode the Device transmits a known data sequence at a programmable bit rate that gives an indication of the error rate on the connection.
This mode can be used for a quick self-test during power-up, but also as a diagnostics tool (to identify unreliable cables or connectors).
Also the connection test mode can be used to verify connection robustness in combination with legacy coax cables (for example when upgrading from analog to digital cameras using existing cables).

4.10 Plug and Play

4.10.1 General

CoaXPress is designed to be plug-and-play.

CoaXPress contains mechanisms for automatic link setup (bit rate, link configuration scheme, Device detection) and Device and Host setup (image format, bit depth, data packing format, etc).

The link is designed to automatically recover after a loss of connectivity.

Comment: These mechanisms make CoaXPress hot-pluggable, and the protocol supports it. However this does not imply that an application will recover if a Device is unplugged and then reconnected. Also there is the potential for damage if a long cable with stored charge is connected.

4.10.2 GenICam

CoaXPress products shall support GenICam as described in section 11.

4.11 Labeling and Use of the CoaXPress Logos

The use of CoaXPress logos on products or marketing material is only allowed for products that have been registered with JIIA. The registered products shall have completed the compliance test procedure and have met the requirements, in accordance with the CoaXPress Compliance Product Certification Program.

Comment: JIIA provides compliance test procedures (section 2.2 ref 8).

4.11.1 Official CoaXPress Logo

The design of the official CoaXPress trademarked logo is shown in the following figure:

Figure 3 — Official CoaXPress Logo

CoaXPress is a registered trademark in Japan, the USA, the EU and other countries and the property of the trademark belongs to Japan Industrial imaging Association (JIIA). You need a trademark license to
use the logo on any product or in commercial documents. If you want to use the trademark or have any questions about using trademarks, please contact a JIIA representative (sec@jiia.org).

### 4.11.2 CoaXPress Logo Varieties

Varieties of CoaXPress logos are shown in the following figure. The first row is the standard color logo, the second row a variant for use on black backgrounds, and the third row the monochrome version. The second column is the short-form logo to be used where space does not permit the full version.

![CoaXPress Logos](image)

*Figure 4 — CoaXPress logos*

Artwork is available from JIIA.

### 4.11.3 Product Labeling

CoaXPress compliant products shall be labeled with the official Logo and indication according to the following table:

<table>
<thead>
<tr>
<th>Maximum operational Bit Rate per Coax (Gbps)</th>
<th>Compliance Labeling</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.250</td>
<td>CXP-1</td>
</tr>
<tr>
<td>2.500</td>
<td>CXP-2</td>
</tr>
<tr>
<td>3.125</td>
<td>CXP-3</td>
</tr>
<tr>
<td>5.000</td>
<td>CXP-5</td>
</tr>
<tr>
<td>6.250</td>
<td>CXP-6</td>
</tr>
</tbody>
</table>
This label indicates that the product meets the CoaXPress specification, with a maximum supported bit rate per coax cable as indicated. It applies to all CoaXPress products, including Hosts, Devices and CXP-Cables.

Cable example: A given length of CXP-Cable marked as \( \text{CXP-5} \) conforms to all the specification requirements for a maximum bit rate of 5 Gbps per coax. It will qualify for the lower bit rates as well. A longer length, but using the same cable type, might only qualify as \( \text{CXP-3} \).

Comment: This labeling is purely indicating CoaXPress compliance. The product vendor is free to label their products to indicate other features in addition to the compliance label.

A Device with more than one coax connector shall label the master connector with an arrowhead \( \rightarrow \) pointing towards the connector.

### 4.11.4 Product Feature Bar

It is recommended that CoaXPress product information (e.g. in brochures, adverts and websites) is labeled with a feature bar in the style shown in the following figure:

![Feature Bar Example](image)

**Figure 5 — Feature bar**

The first box gives the maximum speed, in the format listed in section 4.11.3, and the second box gives the connector type and number of connections. The connector type shall be “BNC” for BNC type connectors, and “DIN” for DIN 1.0/2.3 type connectors. The number of connections shall show the number of downconnections supported, followed where applicable by a “+” sign followed by the number of high speed upconnections supported.

Examples: The left hand feature bar above is for a product with two BNC downlinks running at a maximum of 3.125 Gbps, and the right hand one for a product with DIN 1.0/2.3 connectors, with four downlinks and a high speed uplink all running at a maximum of 6.25 Gbps.

Artwork is available from JIIA.

Comment: This gives a simple “at a glance” guide to help end users and system integrators choose compatible cameras, cables and frame grabbers.
5 Cable and Connector Specification

5.1 Introduction
This section defines CoaXPress cables and connectors from the point of view of a manufacturer of a Device or Host. See “Annex A: Detailed Cable Specification” for detailed cable requirements for cable manufacturers.

CoaXPress uses 75 Ω coaxial ("coax") cables.

A cable compliant with this specification is called a CXP-cable; a bundle of coax cables including connectors is called a CXP-multi-cable. At cable ends one shall have CXP-connectors, or when grouped at ends of a CXP-multi-cable, can have CXP-multi-connectors. The connectors at the Device and the Host side are the receiving jack CXP-connectors and jack CXP-multi-connectors.

Comment: The use of an approved CoaXPress cable is the simplest way to ensure reliable operation, particularly at high speeds and/or long cable lengths. However at lower speeds, and/or with shorter cables, many legacy 75Ω coax cables should operate perfectly.

5.2 Connectors

Comment: The connector standards do not define insertion loss requirements; however this requirement is included in the overall cable and electrical characteristics defined in this specification.

5.2.1 Single Connectors

Connectors shall be of the 75 Ω BNC type, as defined in IEC 61169-8 annex A, or of the 75 Ω DIN 1.0/2.3 type, as defined by the IEC 61169-29 standard. BNC connectors shall also comply with all other requirements in IEC 61169-8 other than the requirement for 75 Ω instead of 50 Ω.

For BNC connectors, that on the Device and Host shall be a socket-center contact type, and that on the cable shall be a plug-center contact type.

For DIN 1.0/2.3 connectors, that on the Device and Host shall be a female type (socket), and that on the cable shall be a male type (plug).
5.2.2 Multi-Connectors

A multiway connector is formed with ‘n’ downconnection channels, plus optionally one high speed upconnection channel. Connection positions for a Device, and dimensions for both Device and Host shall be as shown in the following figure:

![Diagram of multi-connector](image)

**Figure 6 — Device / Host Multi-Connector**

The connections will be in the same order at the Host if a CXP-multi-cable is used.

**Comment:** The master connection and high speed upconnection may need to be routed to the same SERDES in the Host to allow them to both run at the same bitrate, even if other Host connections are routed to different Devices at different bitrates. Therefore some Hosts may only support the high speed upconnection if the master connection is connected to the connector adjacent to the high speed upconnection (which will always be the case if a CXP-multi-cable is in use).

**Comment:** If individual connectors are used on a Host or Device to form a multi-connector, a jig may be needed to achieve the tolerances shown. Note that suitable connectors need to be chosen for the panel thickness used, otherwise the fixing nut may foul the CXP-cable connector.

Connectors used for the multi-connectors shall be of the 75Ω DIN 1.0/2.3 type, as defined by the IEC 61169-29 standard.

The connector on the Device and Host shall be a female type (socket), and that on the cable shall be a male type (plug).
A CXP-multi-cable has a nominally 1mm (min) locking slot in a defined position (see section A.7). Device and Host vendors can supply a bracket that fixes to Device or Host and engages in the slot to provide an additional mechanical locking function by preventing the release tab from being moved. For ease of reference part of the diagram showing the CXP-multi-cable is duplicated here to show the slot:

![Diagram of Multi-Connector Locking Slot](image)

**Figure 7 — Multi-Connector Locking Slot**

### 5.2.3 Couplers

CoaXPress cables can be joined with inline 75 Ω couplers, however these will reduce the usable cable length.

*Comment:* Couplers rated to at least the operational frequency should be used (e.g. 3.125 GHz for 6.25 Gbps), however even the best may reduce cable length by around 5m at 6.25 Gbps.

### 5.2.4 Contact Material

For BNC connectors it is recommended that the center contact is gold plated, and the outer contact either nickel or white bronze plated.

For DIN 1.0/2.3 connectors it is recommended that both the center and outer contacts are gold plated.

*Comment:* These are proven contact materials. Nickel and white bronze (an alloy of copper, tin and zinc) are compatible materials, i.e. they can be used together. Additional approved materials may be added in future revisions of the standard.

### 5.3 Multiple Cables for Connection Aggregation

When more than one physical cable is used to form a link, all the cables shall be nominally the same type and length. The difference in length of the cables shall be less than 1m.

*Comment:* This is to limit the skew in data packets between the multiple cables, and hence to minimize the FIFO requirements at the Host to realign the data. A 1m difference corresponds to approximately a 5ns skew, which is 4 words at 25 Gbps, so easily handled by a FIFO.
5.4 Connector Indicator Lamps

It is recommended that Devices and Hosts have an indicator lamp by each connector. If fitted, these lamps shall include the following indications:

Table 3 — Connector indicator lamp states

<table>
<thead>
<tr>
<th>State</th>
<th>Indication</th>
</tr>
</thead>
<tbody>
<tr>
<td>No power</td>
<td>Off</td>
</tr>
<tr>
<td>System booting</td>
<td>Solid orange</td>
</tr>
<tr>
<td>Powered, but nothing connected (not applicable to a Device reliant on PoCXP power)</td>
<td>Slow pulse red</td>
</tr>
<tr>
<td>Connection detection in progress, PoCXP active</td>
<td>Fast flash alternate green / orange shown for a minimum of 1s even if the connection detection is faster</td>
</tr>
<tr>
<td>Connection detection in progress, PoCXP not in use</td>
<td>Fast flash orange shown for a minimum of 1s even if the connection detection is faster</td>
</tr>
<tr>
<td>Device / Host incompatible, PoCXP active</td>
<td>Slow flash alternate red / green</td>
</tr>
<tr>
<td>Device / Host incompatible, PoCXP not in use</td>
<td>Slow flash alternate red / orange</td>
</tr>
<tr>
<td>PoCXP over-current (Host only)</td>
<td>Solid red</td>
</tr>
<tr>
<td>Device / Host connected, but no data being transferred</td>
<td>Solid green</td>
</tr>
<tr>
<td>Device / Host connected, waiting for event (e.g. trigger, exposure pulse)</td>
<td>Slow pulse orange</td>
</tr>
<tr>
<td>Device / Host connected, data being transferred</td>
<td>Fast flash green</td>
</tr>
<tr>
<td>Error during data transfer (e.g. CRC error, single bit error detected)</td>
<td>500ms red pulse shown for a minimum of 1s even if the connection detection is faster</td>
</tr>
<tr>
<td>Connection test packets being sent</td>
<td>Slow flash alternate green / orange</td>
</tr>
<tr>
<td>Compliance test mode enabled (Device only)</td>
<td>Slow flash alternate red / green / orange</td>
</tr>
<tr>
<td>System error (e.g. internal error)</td>
<td>Fast flash red</td>
</tr>
</tbody>
</table>

Table 4 — Connector indicator lamp timings

<table>
<thead>
<tr>
<th>Indication</th>
<th>Timing (±20%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fast flash</td>
<td>12.5Hz (20ms on, 60ms off)</td>
</tr>
<tr>
<td>Slow flash</td>
<td>0.5Hz (1s on, 1s off)</td>
</tr>
<tr>
<td>Slow pulse</td>
<td>1Hz (200ms on, 800ms off)</td>
</tr>
</tbody>
</table>

It is recommended that both Device and Host have an option to turn off the indicator lamps, for example to allow use in darkened environments. An additional recommended option is to only show error conditions on the lamps.
Examples: If the Device is waiting for trigger, and the Host is ready to accept data, the Device will indicate a slow orange pulse, and the Host solid green. If the triggering was done instead at the Host, with the Device continuously streaming data, the Device would show fast flash green and the Host a slow orange pulse.

Comment: These indications are designed to help a user understand what the system is (or more importantly, isn’t) doing. They will also help Technical Support staff to diagnose customer issues.

Comment: The indications do not try to show the data rate being sent on the connection, only that data is being transferred. Additional indicators can be fitted to show data rate.
6 Electrical Specification

6.1 Introduction

This section defines the electrical parameters from the point of view of a manufacturer of a Device or Host. Reference to datasheets of an approved CoaXPress transceiver (e.g. section 2.2 ref 1) may be useful.

See “Annex B: Chipset Requirements” for detailed requirements for chipset designers (parts of which may also be useful to Device and Host manufacturers as part of product compliance testing).

6.2 Top Level Overview

Figure 8 shows the electrical block diagram of one connection in a CoaXPress system, with the Device on the left linked to the Host on the right by a coax cable with a characteristic impedance of 75 Ω. The figure also defines the test points Tp1 to Tp4 used in this section.

The high speed connection is usually the downconnection from the Device to the Host, and the low speed connection is in the opposite direction, so usually the upconnection from the Host to the Device. In systems using the optional high speed upconnection these are reversed, but for clarity this section describes the usual case. A high speed upconnection shall implement a Host Transceiver in the Device, and a Device Transceiver in the Host according to this section.

Comment: Voltages shown at the output of the PRU and the input to the PTU are typical examples.

The Device has a Phy \( \Phi_D \) (including a high speed serializer and low speed deserializer) connected to a Device Transceiver (DT) that transmits and receives the bidirectional full duplex data over the coax cable. The Device shall use a Device Transceiver compliant with Annex B.

Comment: The Device Transceiver splits the incoming low-speed data signal and the outbound high speed data signal. It also compensates for low frequency loss in the received low speed data, so restoring the original digital bit stream at Tp1.
The I/O pin of the Device Transceiver shall be coupled to the coax cable through a capacitor $C_d$ that AC couples the data between this I/O pin and the center contact of the coax connector.

The Host has a Phy $\Phi_H$ (including a low speed serializer and high speed deserializer) connected to a Host Transceiver (HT) that transmits and receives the bidirectional full duplex data over the coax cable. The Host shall use a Host Transceiver compliant with Annex B.

Comment: The Host Transceiver splits the incoming high-speed data signal and the outbound low speed data signal. It also compensates for frequency dependent attenuation in the coax cable, so restoring the original digital bit stream at $T_{p4}$.

The I/O pin of the Host Transceiver shall be coupled to the coax cable through a capacitor $C_d$ that AC couples the data between this I/O pin and the center connection point of the coax connector.

Connections using Power over CoaXPress (PoCXP) shall also implement a Power Transmit Unit (PTU) or Power Receive Unit (PRU), as described in section 7 on PoCXP.

### 6.3 Capacitor $C_d$

Capacitor $C_d$ at the Host Transceiver side shall have a value of between 80nF and 500nF.

Capacitor $C_d$ at the Device Transceiver side shall have a value of between 25nF and 500nF.

Comment: The 25nF and 80nF limits allow it to couple the low speed data through. A lower capacitor value at the Device side limits the stress on the inputs of the Device Transceiver (DT) when the 24V power supply is switched. The typical value is 100nF at the Host and 33nF at the Device.

The breakdown voltage $V_{BRCD}$ of capacitor $C_d$ at both the Host and the Device side shall be at least 50V.

Comment: This is higher than the PoCXP voltage of 24V, to withstand possible overshoots when the power is ramped up.

### 6.4 Termination

A resistor of 75 $\Omega \pm 15\%$ shall terminate the coax transmission line at both the Host and the Device side.

Comment: This wide tolerance allows the terminators to be implemented within the Device Transceiver (DT) or Host Transceiver (HT), which gives much better high frequency performance than discrete resistors, without violating the return loss requirements (section 6.8). However transceivers using discrete resistors may need to use tighter tolerance terminators, as recommended by the transceiver vendor. At several GHz the termination will not be a pure resistance – this is allowed for in the return loss requirements.

### 6.5 Impedance $Z_p$

Impedance $Z_p$ shall be as high as possible with respect to the 75 $\Omega$ transmission line impedance for the high bit rate signal frequency, and shall be controlled for the low bit rate signal frequency (2 – 10 MHz).

To get a controlled impedance for the low bit rate signal frequency, the inductive part $L_p$ of the impedance $Z_p$ at both the Host and the Device side shall have a value of 11.5 $\mu$H ($\pm 30\%$).

Comment: This can be obtained by placing an inductor of 10 $\mu$H in series with a ferrite bead. In that case, at low bit rate, the inductor is mainly determining the impedance, whilst at high bit rate this inductor...
represents a low impedance due to its parasitic self-capacitance. The ferrite bead in series then makes the impedance high at the high bit rate.

6.6 High Speed Connection

The high speed connection bit rate shall be certain multiples of 625 Mbps starting at 1.250 Gbps as listed in the following table:

<table>
<thead>
<tr>
<th>Supported Bit Rates</th>
<th>Unit Interval (UI)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.250 Gbps</td>
<td>800 ps</td>
</tr>
<tr>
<td>2.500 Gbps</td>
<td>400 ps</td>
</tr>
<tr>
<td>3.125 Gbps</td>
<td>320 ps</td>
</tr>
<tr>
<td>5.000 Gbps</td>
<td>200 ps</td>
</tr>
<tr>
<td>6.250 Gbps</td>
<td>160 ps</td>
</tr>
</tbody>
</table>

The relative tolerance of the bit rate shall be ≤ ±100 ppm.

A Device with multiple connections shall derive the high speed connection data from high speed clocks all derived from one common sub rate master clock, e.g. a 125 MHz clock.

A Device, or a Host implementing a high speed upconnection, shall use a CoaXPress compliant Device Transceiver DT such that the high-speed output waveform is compliant with “Annex B: Chipset Requirements”.

A Host, or a Device implementing a high speed upconnection, shall use a CoaXPress compliant Host Transceiver HT such that the Phy ΦH can reconstruct the original bit stream by compensating for attenuation in the cable as defined in “Annex B: Chipset Requirements”.

The jitter at Tp2 produced by the combination of the Phy $\Phi_D$ and the Device Transceiver (DT) shall not be more than 20% of the Unit Interval (UI). This jitter is shown as $T_j$ in the following eye diagram at Tp2:

![Eye Diagram](image)

**Figure 9 — Definition of the jitter in the transmit EYE at Tp2**

*Comment:* Jitter is defined with respect to the Unit interval, resulting in relaxed normative absolute parameters for lower speed Devices. This jitter will mainly be a result of the clock source to the Phy $\Phi_D$ and the phy itself, but that contributed by the Device Transceiver must be allowed for.

### 6.7 Low Speed Connection

The bit rate of the low-speed connection shall be one sixth of 125 Mbps, being 20.83\(^\circ\) Mbps as listed in Table 6. The relative tolerance of the bit rate shall be $\pm 100$ ppm.

*Comment:* This allows easy clock generation from the 125 Mbps master clock that may be used for generating and locking to the high bit rate clocks. A SERDES function at 20.83\(^\circ\) Mbps is easily implemented in FPGA fabric as a state machine.

<table>
<thead>
<tr>
<th>Supported Bit Rate</th>
<th>Unit Interval (UI)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20.83(^\circ) Mbps</td>
<td>48.000 ns</td>
</tr>
</tbody>
</table>

*Table 6 — Supported low speed connection bit rate*

A Host, or a Device implementing a high speed upconnection, shall use a CoaXPress compliant Host Transceiver HT such that the low-speed output waveform is compliant with “Annex B: Chipset Requirements”.
A Device, or a Host implementing a high speed upconnection, shall use a CoaXPress compliant Device Transceiver DT such that the Phy Φ_d can reconstruct the original bit stream by compensating for attenuation in the cable as defined in “Annex B: Chipset Requirements”.

The jitter at Tp3 produced by the combination of the Phy Φ_h and the Host Transceiver (HT) shall not be more than 5ns. This jitter is shown as $T_{jLF}$ in the following eye diagram at Tp3:

![Eye Diagram](image)

**Figure 10 — Definition of the jitter in the low speed transmit EYE at Tp3**

**Note 1:** For clarity this eye diagram is shown with no simultaneous flow of high speed connection data.

**Note 2:** For clarity this eye diagram is shown with no baseline wandering due to the low-frequency loss from the impedance $Z_p$. (see section B.4.1).
6.8 Return Loss at Connectors

In order to have a measure for the quality of the high frequency front ends of the Device and the Host, the Return Loss is specified at the jack CXP-connectors at Tp2 and Tp3.

Comment: This return loss requirement is achieved by using suitable grade components (including the connector) in the CoaXPress circuitry, and by correct PCB design. See the important guidance notes below.

In order to evaluate whether all this has been done in a way according to this CoaXPress specification, Table 7 and Table 8 define frequency ranges in which the return loss shall be better than 15 dB, 10 dB, 7 dB or 4 dB. These frequency ranges depend on the specified highest bit rate of the connected Device or Host under test. Poor high frequency design (e.g. including stubs, not providing ground plane cut-outs below ferrite beads), or using the wrong inductors or ferrite beads will result in a worse return loss behavior in at least one of the frequency ranges, bringing it out of specification.

Table 7 — Normative return loss frequency ranges for Host

<table>
<thead>
<tr>
<th>Highest bit rate</th>
<th>Frequency range where Return Loss shall be better than -15 dB</th>
<th>Frequency range where Return Loss shall be better than -10 dB</th>
<th>Frequency range where Return Loss shall be better than -7 dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.250 Gbps</td>
<td>5 MHz – 312 MHz</td>
<td>312 MHz – 625 MHz</td>
<td>-</td>
</tr>
<tr>
<td>2.500 Gbps</td>
<td>5 MHz – 625 MHz</td>
<td>625 MHz – 1.25 GHz</td>
<td>-</td>
</tr>
<tr>
<td>3.125 Gbps</td>
<td>5 MHz – 625 MHz</td>
<td>625 MHz – 1.25 GHz</td>
<td>1.25 GHz – 1.62 GHz</td>
</tr>
<tr>
<td>5.000 Gbps</td>
<td>5 MHz – 625 MHz</td>
<td>625 MHz – 1.5 GHz</td>
<td>1.5 GHz – 2.50 Ghz</td>
</tr>
<tr>
<td>6.250 Gbps</td>
<td>5 MHz – 625 MHz</td>
<td>625 MHz – 1.5 GHz</td>
<td>1.5 GHz – 3.2 GHz</td>
</tr>
</tbody>
</table>

Table 8 — Normative return loss frequency ranges for Device

<table>
<thead>
<tr>
<th>Highest bit rate</th>
<th>Frequency range where Return Loss shall be better than -10 dB</th>
<th>Frequency range where Return Loss shall be better than -7 dB</th>
<th>Frequency range where Return Loss shall be better than -4 dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.250 Gbps</td>
<td>5 MHz – 312 MHz</td>
<td>312 MHz – 625 MHz</td>
<td>-</td>
</tr>
<tr>
<td>2.500 Gbps</td>
<td>5 MHz – 312 MHz</td>
<td>312 MHz – 1.25 GHz</td>
<td>-</td>
</tr>
<tr>
<td>3.125 Gbps</td>
<td>5 MHz – 312 MHz</td>
<td>312 MHz – 1.25 GHz</td>
<td>1.25 GHz – 1.62 GHz</td>
</tr>
<tr>
<td>5.000 Gbps</td>
<td>5 MHz – 312 MHz</td>
<td>312 MHz – 1.5 GHz</td>
<td>1.5 GHz – 2.50 Ghz</td>
</tr>
<tr>
<td>6.250 Gbps</td>
<td>5 MHz – 312 MHz</td>
<td>312 MHz – 1.5 MHz</td>
<td>1.5 GHz – 3.2 GHz</td>
</tr>
</tbody>
</table>

Comment: Other industry standards exist that have a more stringent return loss requirement at the highest frequency range. These standards typically allow signal transmission over a series coupling of a cable and a PCB backplane. In these cases stringent return loss is required to control local resonances of data patterns that could destroy signal integrity.

In the CoaXPress standard it is assumed that the HT and DT circuits are located very close to the connector. This relaxes the return loss requirements, allowing easier support of higher bit rates and also to include PoCXP through Zp.
**Important Guidance Notes:**

CoaXPress is likely to be the fastest PCB design that many companies have implemented, therefore some degree of “learning curve” may be needed even though very few components are needed in a typical CoaXPress circuit. The simplest way of meeting these return loss requirements is to **precisely** follow the component and layout recommendations from the manufacturer of the CoaXPress compatible Host Transceiver HT and Device Transceiver DT, and to take advantage of any design review services offered by the manufacturer or their distributors. Note that at multi-Gigabit speeds, using “equivalent” components or small PCB layout changes (even moving a via or changing the physical size of a passive component) can have significant detrimental effects.
7 Power over CoaXPress: PoCXP

7.1 Overview

The available power per cable is 13W, at a nominal 24V. It is anticipated that 13W should be sufficient for most Devices with one CoaXPress coax connector (up to 6.25 Gbps); likewise 26W should be sufficient for most Devices with two connectors (up to 12.5 Gbps), 39W for three connectors etc.

Should more power be needed, a separate power input to the Device can be used, but the intention is that PoCXP is the usual method of powering a CoaXPress Device, so avoiding the higher system cost of a “power brick” for the Device.

Comment: 24V is chosen to allow sufficient power to be available to power most Devices over the long cable lengths that CoaXPress supports. The use of 12V would mean either much lower available power because of IR losses in the cable, or much reduced cable lengths. While the use of 24V means that most frame grabbers will need a 12V to 24V supply, this should cost less much than the power brick that a Device would otherwise need, so minimizing system cost.

PoCXP shall not be implemented on a high speed upconnection.

PoCXP implements Device detection and short circuit protection to minimize the risk of damage should a Device other than a CoaXPress Device be accidentally connected to a CoaXPress Host.

Inductors in both the Host and Device prevent supply noise corrupting image data. Both the transmitter and receiver need to be AC coupled anyway, so they are unaffected by the power signal. The inductors and AC coupling requirements are described in section 6.
7.2 Simplified Example Block Diagram

Figure 11 — PoCXP block diagram

Table 9 — Key to Host block diagram

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Host</strong></td>
<td></td>
</tr>
<tr>
<td>OCP</td>
<td>Over current protection circuit (see section 7.4.3).</td>
</tr>
<tr>
<td>R\textsubscript{i}</td>
<td>Low value sense resistor for current sensing (see section 7.4.4).</td>
</tr>
<tr>
<td>Z\textsubscript{p}</td>
<td>Inductive filter between power and data (see section 6.5).</td>
</tr>
<tr>
<td>C\textsubscript{d}</td>
<td>Data coupling capacitor for the high and low speed connections.</td>
</tr>
</tbody>
</table>

Table 10 — Key to Device block diagram

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Device</strong></td>
<td></td>
</tr>
<tr>
<td>C\textsubscript{d}</td>
<td>Data coupling capacitor for the high and low speed connections.</td>
</tr>
<tr>
<td>Z\textsubscript{p}</td>
<td>Inductive filter between power and data (see section 6.5).</td>
</tr>
<tr>
<td>C\textsubscript{s}</td>
<td>Device input capacitance (see section 7.3.3.2).</td>
</tr>
<tr>
<td>R\textsubscript{s}</td>
<td>Device sense resistance (see section 7.3.3.1).</td>
</tr>
</tbody>
</table>
7.3 Device Requirements

7.3.1 General Requirements

It is highly recommended that all Devices that consume less than 13W per CoaXPress coax connector draw power from the Host via PoCXP (although the Device can have an auxiliary power connector). An exception is that a Device that is inherently self-powered, such as a daisy-chain port on a frame grabber, need not implement PoCXP.

Devices that may consume more than 13W per CoaXPress coax connector shall draw power from an auxiliary connector.

The PoCXP section in a Device is a Power Receiving Unit (see section 6.2).

Comment: Note that the Compliance Test Specification for CoaXPress Devices requires that a sample test camera must be provided that is not powered by PoCXP to allow certain tests including eye diagram checks to be made.

7.3.2 Operating Requirements

7.3.2.1 Voltage

The Device shall operate over a range of 18.5V DC to 26V DC.

Comment: This allows for a 3.5V round-trip drop in the cable compared to the nominal 24V ± 2V output by the Host, corresponding to a maximum round-trip cable resistance of 5Ω. “Round-trip” means the total effect of the center and outer conductors in the coax cable.

The Device shall not be damaged by continuous application of 30V DC and short term startup overshoot up to 50V.

Comment: 30V is the maximum voltage allowable by the “1394 over coax” standard.

7.3.2.2 Power

The Device shall draw a maximum of 13W per cable.

Comment: 13W gives a current of 541mA at the nominal 24V, or 703mA at the minimum 18.5V.

The Device shall not draw more than 50mA until 25ms after its input voltage has reached 15V.

Note: The initial charging current into C_s is excluded from this 50mA figure.

Comment: This defines a startup / brownout requirement, such that the Device does not draw 13W at a voltage much less than 18.5V, and so over-current the Host. The 25ms requirement prevents reflections on the cable causing off-on-off switching, and means that the input voltage should exceed 18.5V by the time the Device PSU starts.

7.3.3 Support for CoaXPress Detection

All CoaXPress Devices that take power from a CoaXPress coax connector shall implement the following requirements. These requirements apply to each connector that takes power.
7.3.3.1 Input Resistance

The Device shall present a sense resistance of $4k\Omega \pm 5\%$ whenever the input voltage at the Device is greater or equal to 2.2V and less than or equal to 5.5V.

Comment: This equates to a sense current of between $446\mu A$ (2.2V, $4k\Omega + 5\%$) and $1.12mA$ (5.5V, $4k\Omega - 5\%$). This wording also allows the sense resistance to be disconnected when the Device is powered at 24V. It also allows the use of an active circuit to implement the resistance, so allowing easy disconnection at 24V. However if left connected, the $4k\Omega$ sense resistance will only result in additional $123mW$ of power dissipation in the Device at 24V. The sense resistance is labeled “$R_s$” in Figure 11.

7.3.3.2 Input Capacitance

The Device shall have a maximum input capacitance $C_s$ on the CoaXPress coax connector of $57\mu F$.

Comment: The value of $57\mu F$ allows a $47\mu F$ 20% component to be used. This capacitor is labeled “$C_s$” in Figure 11.

The Device shall discharge this input capacitance to less than 1V within 500ms of power being removed.

Comment: This allows the Host to correctly sense a camera that gets unplugged and reconnected.

7.3.3.3 Minimum Load Power

After power is applied, the Device shall draw a minimum current of 15mA from the CoaXPress coax connector within 0.25s.

Comment: This allows the Host to detect if the Device is disconnected. 15mA at 24V corresponds to 0.36W.

For avoidance of doubt, this requirement does apply to a Device with more than one CoaXPress coax connector, and consuming more than 13W: It shall draw a minimum of 15mA from a powered connector, even if the other(s) do not yet have power applied.

Comment: The PoCXP state machines in a Host do not know that they are being connected to one camera, so they may be out of sync. Without this requirement it is possible that one connection gets switched off before another one is powered up.

7.3.4 Devices Over 13W

A Device with more than one CoaXPress coax connector, and consuming more than 13W, shall meet the following additional requirements:

The Device’s power supply shall be designed so that it does not draw more than 13W per cable.

Comment: This requirement means that the Device needs to detect when a sufficient number of CoaXPress coax connectors have power and only then start up the Device’s power supply.

The Device’s power supply shall be designed to isolate the CoaXPress coax connectors, i.e. power applied to one connector shall not be injected into any other ones.

7.3.5 Additional Power Connectors

A CoaXPress Device can have an auxiliary power connector to allow it to be used in a system with a separate Device power supply.

Comment: This should only be needed on Devices that consume more than 13W per cable, in which case they would not implement sense resistance $R_s$. 
7.3.5.1 Non-PoCXP CoaXPress Devices

On a Device designed only for use with the auxiliary power connector, it shall not have an input resistance that meets the requirements of section 7.3.3.1. In non-PoCXP CoaXPress Devices the PRU can be omitted.

Comment: Typically the only load provided by a non-PoCXP Device will be one side of the data coupling capacitor $C_d$.

7.3.5.2 Dual Power CoaXPress Devices

A Device can be designed so that it is powered by either PoCXP or by the auxiliary power connector. In this case:

The Device shall be designed to isolate the power sources. In particular the auxiliary connector(s) shall not inject power into the Host, and the CoaXPress coax connector(s) shall not inject power into the auxiliary connector(s).

Comment: This prevents damage in the event that both a PoCXP Host and the separate power supply are used concurrently.

Whenever power is applied to the auxiliary power connector, the Device shall be powered from the auxiliary power connector and shall disconnect the sense resistance $R_s$, such that it shall not have an input resistance that meets the requirements of section 7.3.3.1.

Comment: This requirement prevents a PoCXP Host from continuously applying power to the Device, not sensing the minimum current flowing (see section 7.4.4), and disconnecting power again.

7.4 Host Requirements

7.4.1 General Requirements

All CoaXPress Hosts shall implement PoCXP for each CoaXPress coax connector using a Host Transceiver.

The PoCXP section in a Host is a Power Transmitting Unit (see section 6.2).

7.4.2 Operating Requirements

7.4.2.1 Voltage

When PoCXP is enabled, the Host shall supply 24V DC ± 2V to the CoaXPress coax connector.

Comment: The Host needs to allow for the voltage drop in the OCP, switching circuits, sense circuits and $Z_p$.

7.4.2.2 Power

The Host shall be capable of supplying 17W per cable over the full voltage range defined in section 7.4.2.1.

Comment: This 17W requirement allows for the loss in the cable. A multi-output Host may need to have an auxiliary connector to directly connect to the computer's power supply if the power consumption exceeds that allowed through the bus connector.
7.4.3 Over-Current Protection: OCP

All PoCXP Hosts shall implement an over-current protection circuit with the following characteristics:

- A holding current over the Host’s operating temperature range of at least 790mA.
- A nominal trip current over the Host’s operating temperature range of not more than 5A.

It is recommended that a UL approved component is used to implement the OCP.

Comment: The OCP prevents serious damage or fire risk in the event of a failure in the Device, damage to the cable, or accidental coupling to a non-CoaXPress Device that happens to satisfy the PoCXP detection system. A PTC resettable fuse should satisfy this requirement providing its operating voltage is at least 26V.

7.4.4 Support for CoaXPress Detection

All CoaXPress Hosts shall implement the following requirements.

These requirements apply to each connector.

- The Host shall sense that a PoCXP Device is connected before applying power. It can do this by sensing the Device’s nominal 4k7Ω input resistance “R_in”, in which case it shall use a sense current of between 550μA and 1mA.
  
  Comment: This gives a sensed voltage of 2.46V to 4.94V after allowing for the RC time constant of 0.28s from the 57μF maximum capacitance “C_s” with the 4k7Ω input resistance. The sense current range of the Host is deliberately narrower than that implied by section 7.3.3.1 to give a noise margin to the system.

- Once power has been applied to a PoCXP Device, the Host shall monitor the power supply current consumed. If at any time after the initial 0.3s the average current is less than 8mA for 0.5s, the Host shall remove power from the Device.
  
  Comment: This causes power to be removed if the Device or cable is unplugged. It also allows the state machine controlling PoCXP to return to its “Sense” state ready to detect a new Device. The current sensing circuit can also be used to monitor the actual power consumed by the Device (and cable), and switch off power should the current exceed certain limits, but this is not a requirement of the specification. The figure of 8mA for the Host, compared to 15mA for the Device, gives a noise margin to the system.

- Once power has been applied to a PoCXP Device, the Host shall monitor the voltage on the CoaXPress coax connector. If for more than 20ms this voltage drops to a level indicating that the OCP has tripped, the Host shall remove power from the Device.
  
  The Host shall specifically be designed not to apply power to the following non-PoCXP Devices:
  
  - Non-PoCXP CoaXPress Device, or analog Device, where the Device has a coupling capacitor as the only load.
  - 50Ω or 75Ω load to ground.
  - Short circuit to ground.
  - Large capacitative load such that the sensed voltage is not tending towards a stable value after 3 RC time constants of 4k7Ω / 57μF.
The Host shall not exceed the following values while sensing for a PoCXP Device:
- Maximum voltage with no load connected: 7V.
- Maximum short circuit current: 2mA.

The following figure gives a simplified example state diagram for PoCXP Device detection:

**Figure 12 — Simplified example state diagram**

### 7.4.5 Additional Features

It is recommended that the Host supports a mode to allow power to be disconnected from the Device under user control, e.g. to enable standby modes, or to perform a “hard reset” on the Device.

Similarly it is recommended that the Host supports a mode to allow power to be re-connected to the Device, by forcing the Device to re-detect if a PoCXP Device is connected.

*Comment: Note that the Host cannot support a mode where it always applies power, as this would violate section 7.4.4.*
8 Link Protocol

8.1 Overview

The link protocol defines a point to point interface between a Host and Device. The link consists of a Master physical connection and optional extension connections and high speed upconnection.

All data is transferred using 8B/10B coded packets.

For each connection the protocol defines a set of logical channels carrying specific data such as stream data (e.g. images), real time triggers and Device control.

The Device is controlled by the Host via registers. Host register access is provided via a control channel.

When no high speed upconnection is in use, write access and therefore control is exclusively provided via the master connection making the connected Host connection the control master. Extension connections provide read-only access allowing Device and connection discovery. When the high speed upconnection is in use it replaces the master low speed upconnection to provide Device control.

The logical channels when no high speed upconnection is in use are shown in the following figure:

![Link Overview Diagram]

* = Discovery only

---

Figure 13 — Link Overview
The logical channels when the high speed upconnection is in use are shown in the following figure:

**Figure 14 — Link Overview with High Speed Upconnection**

Comment: These figures are best viewed in color.
8.2 Transport Layer

8.2.1 Packet Transmission Format

Both the upconnection and downconnection shall use 8B/10B coding as defined in section 2.1. Besides the 256 data characters D0.0 – D31.7, K-codes are used for formatting and indication purposes. An overview of K-code usage is listed in the following table:

<table>
<thead>
<tr>
<th>K-code</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>K27.7</td>
<td>Start of data packet indication</td>
</tr>
<tr>
<td>K28.6</td>
<td>I/O acknowledgment</td>
</tr>
<tr>
<td>K28.1</td>
<td>Used for alignment</td>
</tr>
<tr>
<td>K28.2</td>
<td>Trigger indication</td>
</tr>
<tr>
<td>K28.3</td>
<td>Stream marker – see section 9.2</td>
</tr>
<tr>
<td>K28.4</td>
<td>Trigger indication</td>
</tr>
<tr>
<td>K28.5</td>
<td>Used for alignment</td>
</tr>
<tr>
<td>K29.7</td>
<td>End of data packet indication</td>
</tr>
</tbody>
</table>

Transmission is organized in groups of 4 characters called a word, except for low speed connection trigger packets which are six characters.

A word consists of 4 consecutive 8B/10B characters labeled P0, P1, P2 and P3 and are transmitted in the stated order (i.e. P0 first). The receiver shall perform word alignment in order to successfully decode the packets.

The bit transmission order shall be according to the 8B/10B coding scheme: 8 bits labeled { ‘A’, ‘B’, ‘C’, ‘D’, ‘E’, ‘F’, ‘G’, ‘H’ } and a data/control selection are coded into a 10 bit symbol with bits labeled { ‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘i’, ‘f’, ‘g’, ‘h’, ‘j’ }. Prior to coding the bit labeled ‘A’ is the LSB. After coding the bit labeled ‘a’ is transmitted first. This order is shown in the following figure:

```
To connection
0011111010 1100000110 0011111001 1010101010
```

Transmission is organized in groups of 4 characters called a word, except for low speed connection trigger packets which are six characters.

A word consists of 4 consecutive 8B/10B characters labeled P0, P1, P2 and P3 and are transmitted in the stated order (i.e. P0 first). The receiver shall perform word alignment in order to successfully decode the packets.

The bit transmission order shall be according to the 8B/10B coding scheme: 8 bits labeled { ‘A’, ‘B’, ‘C’, ‘D’, ‘E’, ‘F’, ‘G’, ‘H’ } and a data/control selection are coded into a 10 bit symbol with bits labeled { ‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘i’, ‘f’, ‘g’, ‘h’, ‘j’ }. Prior to coding the bit labeled ‘A’ is the LSB. After coding the bit labeled ‘a’ is transmitted first. This order is shown in the following figure:

```
To connection
0011111010 1100000110 0011111001 1010101010
```

Unique single values represented by multiple bytes (e.g. a 32 bit address) shall be transmitted using big endian byte order.

---

CoaXPress Standard Version 1.1.1 33 JIIA CXP-001-2015
8.2.2 Bit Error Handling

The link protocol (i.e. packet structure, overall control) is designed to be immune to single bit errors by a combination of careful choice of packet coding, and sending key information multiple times.

Any errors in data transfer are detectable by a CRC, but this revision of the protocol does not support stream data resend.

8.2.2.1 Duplicated Characters

Many words duplicate the same value in P0, P1, P2 and P3. The receiver shall decode these multiple packets in a way that provides immunity to single bit errors.

Comment: In general information is sent four times rather than the usual three times. This is simply to maintain word alignment, and three of the characters are sufficient to “vote” and cope with single bit errors. However having four duplicates may allow sophisticated receivers to decode the packet correctly in the event of multiple bit errors, but that is a quality of implementation decision beyond the requirements of the specification.

8.2.2.2 CRC

A CRC is used to protect stream data packets and control packets.

For control packets the receiver shall decode these CRCs and in the event of a CRC error shall send a negative acknowledgment indicating the error.

Comment: How this acknowledgment is handled is beyond the scope of this specification, but it is recommended that the command is resent up to three times, and if the third retry fails an error message would be passed up to the application to deal with.

For stream data packets the receiver shall decode these CRCs for error monitoring purposes.

Comment: How CRC errors are handled is beyond the scope of this specification, but typically an error message would be passed up to the application to deal with, either after every error, or after the number of errors exceeds a threshold.

All CRCs defined in the link protocol are 32 bit CRCs calculated over a specified block of data.

Polynomial: \[ x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^9 + x^7 + x^5 + x^4 + x^2 + x + 1 \]

Seed: 0xFFFFFFFF

Comment: This is the CRC-32 polynomial defined by IEEE-802.3 (Ethernet). Note that CoaXPress does not use the CRC as defined by IEEE-802.3 – the CRC calculation is defined below.

The CRC shall be calculated over the data using the following bit order:

- Data bit 0 (lsb) first within a character; bit 7 (msb) last.
- Character P0 first within a word; character P3 last.
- Word 0 first within a packet; word \( N \) last, where \( N \) is the number of words over which the CRC needs to be calculated (see individual packet definitions).

The calculated CRC shall be sent over the connection in the following bit order: The MSB of the CRC sent in character P0 bit 0 through to the LSB in character P3 bit 7.

Comment: This ensures that the MSB is sent first over the serial connection, to correctly give a zero value in the receiver CRC register after both data and the transmitted CRC have been processed.
The bit order is shown in the following figure:

```
<table>
<thead>
<tr>
<th>P0</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
</tr>
</thead>
<tbody>
<tr>
<td>00111101</td>
<td>00111100</td>
<td>00111100</td>
<td>10101101</td>
</tr>
</tbody>
</table>
```

First bit inserted

Data Word

```
00111101
P0
00111100 00111100
P1 P2
10101101
P3
```

First bit inserted

Swap

```
BC 3C 3C B5
```

e.g. IDLE word

```
00111101
P0
00111100 00111100
P1 P2
10101101
P3
```

First bit inserted

Swap

```
P0 P1 P2 P3
```

Resulting CRC

Comment: For clarity, this shows a serial implementation.

The CRC shall not include idle words used to stretch data transmission. The K28.3 stream marker shall be treated as a D28.3 for CRC calculation, i.e. the K-Code flag is ignored.

Comment: To aid understanding, a complete control command packet (a read of address 0) is shown here, with the resulting CRC shown in red: K27.7 K27.7 K27.7 K27.7 0x02 0x02 0x02 0x02 0x00 0x00 0x00 0x00 0x04 0x00 0x00 0x00 0x00 0x56 0x86 0x5D 0x6F K29.7 K29.7 K29.7 K29.7.
8.2.3 Packet Types

Information is transferred in both directions using packets. The start of a packet is indicated by a word containing K-codes. Two main packet types are used:

- Dedicated short packets for real time information are transmitted as a specific start indication (one of K28.0, K28.2 or K28.4), followed by the fixed sized data content.
- A more generic data packet consisting of a start indication (K27.7), the packet type and an end indication (K29.7).

Packet types defined by the link protocol are listed in the following table:

<table>
<thead>
<tr>
<th>Channel</th>
<th>Packet Type</th>
<th>Usage Downconnection</th>
<th>Usage Upconnection</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/O</td>
<td>Trigger packets</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Stream</td>
<td>Data packet, type = { Stream }</td>
<td>✓</td>
<td>-</td>
</tr>
<tr>
<td>Control</td>
<td>Data packet, type = { Control data, Connection Test }</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Comment: In principle stream data could be sent on the up connection, but the current discovery procedure does not allow for this, and the specification is written with only the Device generating stream data.

8.2.4 Packet Transmission Order

With the exception of trigger packets, and their associated acknowledgements, Devices and Hosts are free to prioritise sending packets to optimize performance.

Comment: A Device may choose to insert a small control packet acknowledgement as soon as the next image data packet finishes, to allow good GUI response in a user application. However if the control packet was to read a large data block, and its image data FIFOs were nearly full, it may continue to send image data and delay the control acknowledge.

Note: Section 8.5.3 requires packets within one stream to be transmitted in order.

To maintain guaranteed trigger timing, trigger packets and trigger acknowledges have a defined higher priority. A high priority packet shall be inserted into a lower priority packet that is being transmitted. The insertion shall take place at a word boundary with an exception for the low speed connection trigger packet which shall take place at a character boundary.

Comment: Allowing character insertion of the low speed trigger packet allows minimal trigger latency given the relatively low speed of the connection.

Transmission of the lower priority packet shall be resumed after the transmission of the inserted packet is completed.
The link protocol defines 3 levels of priority as indicated in the following table:

### Table 13 — Packet transmission priority

<table>
<thead>
<tr>
<th>Priority Level</th>
<th>Packet type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (highest)</td>
<td>Trigger</td>
</tr>
<tr>
<td>1</td>
<td>I/O acknowledgment (for trigger packet)</td>
</tr>
<tr>
<td>2 (lowest)</td>
<td>All other packets</td>
</tr>
</tbody>
</table>

An example of priority based transmission on the high speed connection is illustrated in the following figure for a trigger packet being inserted into a data packet:

**Explanation of terms used:**
- \( p\)-data = Packet data.
- \( t\)-data = Trigger timing data.

**Figure 17 — High speed connection packet transmission example**
The following figure shows the same sequence on the low speed connection, where the trigger packet is inserted on a character boundary:

![Diagram showing low speed connection packet transmission example]

**Figure 18 — Low speed connection packet transmission example**

The following figure shows an additional example on the low speed connection, where rising and falling triggers are inserted on a character boundary:

![Diagram showing low speed connection trigger insertion example]

**Figure 19 — Low speed connection trigger insertion example**
8.2.5  Idle Transmission

A connection is idle when no packet transmission takes place. This idle time shall be filled with IDLE words. The format of this word is defined in the following table:

<table>
<thead>
<tr>
<th>Word</th>
<th>P0</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDLE</td>
<td>K28.5</td>
<td>K28.1</td>
<td>K28.1</td>
<td>D21.5</td>
</tr>
</tbody>
</table>

Comment: The idle word serves the following purposes at the receiver:
- Allows rapid bit clock lock by means of the high frequency (D21.5) data content.
- Provides low frequency (K28.5, K28.1) and high frequency (D21.5) data content to serve the line equalizer.
- Allows character and word alignment using the K28.5 character that is only used as character P0 in the idle word.
- Allows monitoring of the alignment status once alignment is achieved.

8.2.5.1  Idle Usage

Both high and low speed connection transmitters shall send the IDLE word when the connection is idle. For the avoidance of doubt, low speed connection transmitters shall continue to send the IDLE word on extension connections, and on the master connection if the optional high speed upconnection is in use.

For proper operation of a high speed connection an IDLE word shall be transmitted at least once every 100 words.

For proper operation of a low speed connection an IDLE word shall be transmitted at least once every 10,000 words.

8.2.5.2  Stretching Packet Transmission with IDLE

The transmission of a packet on the high speed connection can be stretched by inserting IDLE words. These IDLE words should be ignored by the receiver and are not included in the CRC.

Comment: The possibility to insert IDLE words relaxes transmitter buffer requirements – i.e. the entire packet does not need to be available when the packet transmission commences. This could be useful with control channel reads. However for data stream packets it is expected that the entire packet would be buffered, so no additional IDLE words would need to be inserted.

Additional IDLE words to stretch packets shall not be inserted into a packet on the low speed connection.

Comment: There is no need to stretch low speed connection packets because of the slow data rate, and banning this simplifies implementation.
8.3 I/O Channel

The I/O channel provides trigger in 2 directions. The I/O channel is defined for the Master connection (connection 0) only.

8.3.1 GPIO

IMPORTANT: GPIO has been removed from v1.1. It is planned to replace GPIO with a new event packet structure using K28.0 in v1.2 of the specification. GPIO will simply become one event type.

8.3.2 Trigger

Trigger is provided in 2 directions: From Host to Device, and from Device to Host.

Triggers have the highest priority and shall be inserted into any lower priority transmission packet at the appropriate transmission boundary. To guarantee proper initialization both the Host and Device shall de-assert the trigger signal as part of link discovery, which shall correspond to the effect of a falling edge trigger packet.

Trigger packets shall be acknowledged (see section 8.3.3).

8.3.2.1 Low Speed Connection Trigger

The event that results in a trigger being sent could occur at any time, but the trigger packet cannot be sent until the next transmission boundary. For the low speed connection the trigger packet can start at the next character boundary (see section 8.2.4).

Comment: Allowing character alignment of the low speed trigger packet allows minimal trigger latency given the relatively low speed of the connection.

To minimize trigger jitter, the time between the trigger event and the trigger packet being sent is coded into the packet as a delay value. The receiver can then use this delay to recreate the trigger event with low jitter and a fixed latency.

For the low speed connection the delay value is 239 minus the number of units of 1/24 the low speed connection bit interval (2.00 ns) between the trigger event and the start of the trigger packet. See Figure 20.

Comment: Transmission timing accuracy is limited to character units. A character contains 10 bit intervals (after 8B/10B coding).

A bit interval equals 1 / 20.83 MHz = 48 ns. The character transmission time is therefore 10 * 48 ns.

The trigger timing correction delay value is coded in units of 2 ns. A full character interval is therefore 480 / 2 = 240 delay value codes wide.

Therefore to encode the trigger delay correction value, a range of 0 to 239 is needed.
Figure 20 — Low speed trigger example
Table 15 — Low speed connection trigger packet format

<table>
<thead>
<tr>
<th>Char</th>
<th>Content</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 – 2</td>
<td>Either K28.2</td>
<td>Trigger packet indication – rising edge</td>
</tr>
<tr>
<td>0 – 2</td>
<td>K28.4 K28.4</td>
<td>Trigger packet indication – rising edge</td>
</tr>
<tr>
<td>or</td>
<td>K28.4 K28.2</td>
<td>Trigger packet indication – falling edge</td>
</tr>
<tr>
<td>3 – 5</td>
<td>3x Delay</td>
<td>The delay value is 239 minus the number of units of 1/24 the low speed connection bit interval (2.00 ns) between the trigger event and the start of the trigger packet. The Host can use a coarser time unit by setting lower bits of this value fixed at 0 or giving it a bias. The Device can use a coarser time unit by using less bits of this value (discard non used lower bits of this value or use rounding). The chosen accuracy and usage of this timing correction value is left as a quality of implementation at both the Host and Device.</td>
</tr>
</tbody>
</table>

Packet size: 6 characters (6 bytes).

8.3.2.2 High Speed Connection Trigger

Similarly for the high speed connection the next transmission boundary is the next word (4 characters), and the delay value is three minus the number of whole characters between the trigger event and the start of the trigger packet.

Table 16 — High speed connection trigger packet format

<table>
<thead>
<tr>
<th>Word</th>
<th>Content</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Either 4x K28.4</td>
<td>Trigger packet indication – rising edge</td>
</tr>
<tr>
<td>or</td>
<td>4x K28.2</td>
<td>Trigger packet indication – falling edge</td>
</tr>
<tr>
<td>1</td>
<td>4x Delay</td>
<td>The delay value is three minus the number of whole characters between the trigger event and the start of this trigger packet. Usage is optional, when not used set to 0.</td>
</tr>
</tbody>
</table>

Packet size: 2 words (8 bytes).
8.3.3 I/O Acknowledgments

Trigger packets shall be acknowledged with an I/O acknowledgment packet.

Table 17 — I/O acknowledgment packet format

<table>
<thead>
<tr>
<th>Word</th>
<th>Content</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4x K28.6</td>
<td>I/O acknowledgment packet indication</td>
</tr>
<tr>
<td>1</td>
<td>4x Code</td>
<td>Acknowledgment code, repeated 4 times: 0x01 Trigger packet received OK</td>
</tr>
</tbody>
</table>

Packet size: 2 words (8 bytes).

The Device or Host transmitting a trigger packet shall use the following rules:

- After completion of a trigger packet transmission it shall not send a new trigger packet until it has received an acknowledgment from the Host or Device receiving the packet.
- However if it does not receive an acknowledgment for more than a defined time (transmission timeout, see below) it can resend the last trigger packet or can send a new trigger packet.

Transmission timeout rules:

- Trigger packet sent on low speed connection, acknowledgment on high speed connection (therefore I/O is from Host to Device): The timeout shall be one character on the low speed connection.
- Trigger packet sent on high speed connection, acknowledgment on low speed connection (therefore I/O is from Device to Host): The time to send the acknowledgment depends on the usage on the low speed connection (see comment below). Therefore it is recommended that the Device has a register controlling the timeout, set to a default value by the Device based on its I/O usage, but that can be overridden by the Host.
- Trigger packet sent on high speed connection, acknowledgment on high speed connection (therefore high speed upconnection in use): The timeout shall be 480ns.

Comment: The decision on what to do in the event of a trigger packet being lost is system-dependent, and likely to be controlled by the Host. A Device may need a register that allows the Host to determine what it does.

Comment: Note that low speed connection has limited bandwidth, and the system designer needs to choose trigger rates in both directions such that the bandwidth is not exceeded. For instance if it is necessary to run the Host to Device trigger at near to the maximum capacity (e.g. for a line trigger on a line scan device), then the Device should not send triggers to the Host, as these would require trigger acknowledgments which there would not be time to send. This situation would be detected by transmission timeouts in the Device. Of course this is not a limitation when using a high speed upconnection which dedicates its bandwidth to the transmission of triggers.
8.4 Data Packet Definition

Table 18 — Data packet transmission format

<table>
<thead>
<tr>
<th>Word</th>
<th>Content</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4x K27.7</td>
<td>Start of packet indication</td>
</tr>
<tr>
<td>1</td>
<td>4x Type</td>
<td>Data packet type:</td>
</tr>
<tr>
<td></td>
<td>0x00</td>
<td>Reserved for extended types</td>
</tr>
<tr>
<td></td>
<td>0x01</td>
<td>Indicates a stream data packet</td>
</tr>
<tr>
<td></td>
<td>0x02</td>
<td>Indicates control command</td>
</tr>
<tr>
<td></td>
<td>0x03</td>
<td>Indicates control acknowledge</td>
</tr>
<tr>
<td></td>
<td>0x04</td>
<td>Indicates a connection test packet</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Other values are reserved for future use.</td>
</tr>
<tr>
<td>2…</td>
<td>Data (4 bytes)</td>
<td>Data packet payload of D data words</td>
</tr>
<tr>
<td>D+1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D+2</td>
<td>4x K29.7</td>
<td>End of packet indication</td>
</tr>
</tbody>
</table>

Packet size: D + 3 words ((D * 4) + 12 bytes).

The following sections define these data packet payloads.
8.5 Stream Data Packet

Stream data packets are used to send stream data (see section 9.1) across the link. The packet transfer layer is responsible for forming packets from a stream by chopping the stream into blocks and adding a stream packet header and trailer to form a packet. This is shown diagrammatically in the following figure:

```
<table>
<thead>
<tr>
<th>Line 1</th>
<th>Line 2</th>
<th>Line 3</th>
<th>Line 4</th>
<th>Line 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw line data</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

- **Form into a stream**
  - Raw line data
  - LM
  - LM
  - LM
  - LM
  - IH
  - IH
  - IH
  - IH
  - Pkt 0
  - Pkt 1
  - Pkt 2
  - Pkt 3
  - Pkt 4
  - Pkt 5

- **Packetize**
  - SPH
  - SPH
  - SPH
  - SPH
  - SPT
  - SPT
  - SPT
  - SPT
  - SPT
  - SPT

**Stream Markers:**
- IH = Image Header
- LM = Line Marker
- SPH = Stream Packet Header
- SPT = Stream Packet Trailer

Figure 21 — Packet formation in Device – image data example

Comment: This figure is best viewed in color.
The packet transfer layer is also responsible for sharing all the packets from the Device across the available connections, as described in the following sections.

### 8.5.1 Stream Data Packet Format

The packet format is shown in the following table:

<table>
<thead>
<tr>
<th>Word</th>
<th>Content</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4x K27.7</td>
<td>Start of packet indication (see section 8.4).</td>
<td></td>
</tr>
<tr>
<td>4x 0x01</td>
<td>Stream data packet indication.</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>4x Stream ID</td>
<td>Unique stream ID that this packet contains data for.</td>
</tr>
<tr>
<td>1</td>
<td>4x Packet Tag</td>
<td>8 bit tag. Incremented for each packet with this stream ID, wraparound to 0 at 0xFF.</td>
</tr>
<tr>
<td>2</td>
<td>4x DsizeP(15:8)</td>
<td>16 bit value representing the number of stream data words N per packet.</td>
</tr>
<tr>
<td>3</td>
<td>4x DsizeP(7:0)</td>
<td></td>
</tr>
<tr>
<td>4 to N+3</td>
<td>Stream data</td>
<td>N words of stream data.</td>
</tr>
<tr>
<td>N+4</td>
<td>CRC</td>
<td>32-bit CRC calculated over the stream data 4 to (N+3).</td>
</tr>
<tr>
<td>4x K29.7</td>
<td>End of packet indication.</td>
<td></td>
</tr>
</tbody>
</table>

Packet size: $N + 8$ words ($(N \times 4) + 32$ bytes).

Note: The Stream Packet Header (SPH) and Stream Packet Trailer (SPT) are shown in the right hand column.

**Comment:** The Stream ID is the same as in the image header (e.g. section 9.4.6.2). This allows the packet decoder in the Host to read the Stream ID and therefore send the packet's contents to the appropriate buffer for that stream. The Stream ID will still be in that buffer (in the image header), and will be needed to process the stream.

### 8.5.2 Packet Size

The total stream packet size (i.e. from the first K27.7 to the last K29.7) shall not be more than the size set in `StreamPacketSizeMax` during Device discovery (see section 10.1.5).

The Device can use any packet size it wants to up to this size.

**Comment:** The largest supported packet size is recommended for maximum efficiency. A packet size of around 1 Kbytes gives a low packet overhead combined with practical buffer sizes in FPGAs. Depending on the image size compared to the packet size, note that the last data packet in an image could be very small – potentially only transferring 1 word. Also note that bootstrap register `StreamPacketSizeMax` is set to zero during Device discovery; therefore the Device cannot send any packets until Device discovery completes and the register is set to the correct value.
8.5.3 Packet Order and Packet Tag

Packets within one stream shall be transmitted in order, i.e. the first packet in a stream shall have packet tag 0, then the data immediately following shall be the next packet with tag 1, then tag 2 etc.

Comment: The tag allows the Host to check for missing packets, and to confirm the correct packet order when multiple connections are in use. Given that the physical layer is deterministic, the protocol does not need to allow for packets arriving out of order.

The packet tag shall only be reset back to zero as part of a ConnectionReset (see section 10.3.28) or a write to ConnectionConfig (see section 10.3.33).

Comment: Therefore the packet tag is not reset for example by an AcquisitionStart or AcquisitionStop, a start or end of a frame, or a change of image size.

8.5.4 Combining Multiple Streams

After completing one packet transmission, the Device can send the next packet from any stream with available data.

It is the responsibility of the Device to prioritize packet transmission order based on its available buffer sizes.

Comment: A simple Device may only have one stream, so does not need this process.

This is shown in the following figure:

Figure 22 — Packet processing in Device
8.5.5 Packet Transfer over Multiple Connections

Successive packets in a stream, or a combined stream (see section 8.5.4), shall be written to the next connection in ascending order of Connection ID, starting at connection 0, as shown in the following table. Also see Figure 22.

Table 20 — Packet order on connections – example

<table>
<thead>
<tr>
<th>Connection 0</th>
<th>Connection 1</th>
<th>Connection 2</th>
<th>Connection 3</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Packet 0</td>
<td>Packet 1</td>
<td>Packet 2</td>
<td>Packet 3</td>
<td></td>
</tr>
<tr>
<td>Packet 4</td>
<td>Packet 5</td>
<td>Packet 6</td>
<td>Packet 7</td>
<td></td>
</tr>
<tr>
<td>Packet 8</td>
<td>Packet 9</td>
<td>Packet 10</td>
<td>Packet 11</td>
<td></td>
</tr>
<tr>
<td>Packet 12</td>
<td>Packet 13</td>
<td>Packet 14</td>
<td>Packet 15</td>
<td></td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

Note that the Host knows the connection order from the bootstrap registers DeviceConnectionID it reads during Device discovery (see section 10.1.3).

Comment: A simple Device may only have one connection, so does not need this process. The defined order means that the Host knows which connection the next packet will be on, without needing to read the packet header to find out.

The connection order shall only be reset to start at connection 0 as part of a ConnectionReset (see section 10.3.28) or a write to ConnectionConfig (see section 10.3.33).

Comment: Therefore the connection order is not reset for example by an AcquisitionStart or AcquisitionStop, a start or end of a frame, or a change of image size.

8.6 Control Channel

The control channel is used to provide register access through specific control commands, transmitted from Host to Device, and the resulting acknowledgment messages, transmitted from the Device to the Host.

The following command messages are defined:

- Memory read of \( B \) bytes.
- Memory write of \( B \) bytes.
- Control channel reset.

Acknowledgment messages contain an acknowledgment code. This code indicates:

- Success, meaning that the command executed (or is still executing) OK, and in the case of a memory read, the requested data is attached.
- Logical Errors, meaning that the command packet was received OK but attempted an invalid operation.
- Physical Errors, meaning that the command packet was corrupt.

Comment: The processing of logical and physical errors by the Host, and hence the application, may be very different.
Control commands shall only be sent by the Host. Acknowledgment messages shall only be sent by the Device as a response to a received control command.

The master connection control channel shall provide full register access while extension connection control channels shall be restricted to read-only register access. Read access via an extension connection control channel shall only be utilized during Device discovery.

*Comment:* The control tasks defined in the current specification are limited to Device register access by the Host.

### 8.6.1 Control Transactions

A control transaction is composed of transmitted command, the execution of that command, and finally an acknowledgment as shown schematically in the following figure:

![Figure 23 — Control transaction – standard](image)

If the transaction will take longer than the allowed timeout (200ms, defined in section 8.6.1.1) a wait acknowledgment can be sent to specify the required time (maximum 10 seconds, defined in section 8.6.3), as shown schematically in the following figure:

![Figure 24 — Control transaction – wait acknowledgment](image)

*Comment:* If a Device needs more than 10 seconds for a certain action, or does not know how long an action will take, it can use one control transaction to start the action, and then poll a separate register to check for completion. A GenICam command supports this method, and suitable XML code can specify the polling register and if necessary disable other Device features until the command has completed.

---

2 There is one exception: When the connection speed is changed by writing to `ConnectionConfig`, the Device is required to acknowledge the write before changing the speed. See section 10.3.33.
8.6.1.1 Transaction Rules

The Host shall implement the following rules for control transactions:

- After completion of command transmission the Host shall not send a new command until it has received a final acknowledgment from the Device.
- However the Host can resend the last command or can send a new command:
  - If the Host does not receive an acknowledgment (final acknowledgment or wait acknowledgment) for more than 200ms (transmission timeout), or
  - If following a wait acknowledgment the Host does not receive a final acknowledgment within the time given in the wait acknowledgement, or
  - The Host can send a control channel reset before receiving a final acknowledgment. See section 8.6.1.2.

The Device shall implement the following rules for control transactions:

- On receiving a valid command the Device shall execute the command and transmit a final acknowledgment after command execution.
- On receiving an invalid command the appropriate acknowledgment shall be transmitted immediately and the command shall be discarded.
- Command execution time and the time required to transmit the final acknowledgment shall not exceed 200ms, which is the transmission timeout.
- However if the Device needs more than 200ms it shall transmit one wait acknowledgment within 200ms to inform the Host that it is processing the last command. The wait acknowledgment shall specify the maximum time that the Host should wait for the final acknowledgement after receiving the wait acknowledgment. The Device shall then send one final acknowledgment within the specified time.

Comment: Wait acknowledgments are not allowed for accesses to bootstrap registers. See section 10.3.3.

8.6.1.2 Control Channel Reset

The Host can transmit a control channel reset message to try to regain control of a Device that is not responding correctly. On receiving this message, the Device shall abort any control operation in process, reset its control channel logic, and send the unique success message.

Comment: Control channel reset needs to be used with care. Resetting the Device part way through (for instance) a flash memory update could render the Device unusable.

If the Device's control channel is not working, it is quite possible that it will not respond to the control channel reset message, and the only recovery possible may be to power cycle the Device. See section 7.4.5.

Note that control channel reset is not a Device reset – it shall only reset the control channel, not the entire Device.
8.6.2 Control Command Message Format

Control commands are transmitted in the payload section of a data packet with packet type 0x02. The payload format is specified in the following table:

<table>
<thead>
<tr>
<th>Word</th>
<th>Content</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4x K27.7</td>
<td>Start of packet indication</td>
<td>(see section 8.4).</td>
</tr>
<tr>
<td>4x 0x02</td>
<td>Control command indication.</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Cmd</td>
<td>Control operation code (in character P0):</td>
</tr>
<tr>
<td></td>
<td>0x00</td>
<td>Memory read</td>
</tr>
<tr>
<td></td>
<td>0x01</td>
<td>Memory write</td>
</tr>
<tr>
<td></td>
<td>0xFF</td>
<td>Control channel reset</td>
</tr>
<tr>
<td></td>
<td>Others values are reserved</td>
<td>for future use.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Size</td>
<td>24 bit value, stored in</td>
<td>consisting of 3 characters P1,P2,P3, that specifies the number of data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>bytes B to be read or written. B does not include any dummy bytes for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>padding.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This value shall be ≥ 1 (at least 1 byte to read/write) except for a control</td>
</tr>
<tr>
<td></td>
<td></td>
<td>channel reset when it shall be 0.</td>
</tr>
<tr>
<td>1</td>
<td>Addr</td>
<td>32 bit address of the memory location to read from / write to.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When more than one byte must be read / written the successive addresses</td>
</tr>
<tr>
<td></td>
<td></td>
<td>are in increasing order starting from this address.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Set to 0x00000000 for a control channel reset.</td>
</tr>
<tr>
<td>2 to</td>
<td>Write</td>
<td>Data to be written.</td>
</tr>
<tr>
<td>(N-1)+2</td>
<td>Data</td>
<td>This field is omitted for a read operation (Cmd=0x00) or a control channel</td>
</tr>
<tr>
<td></td>
<td></td>
<td>reset (Cmd=0xFF).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The size of this data section shall be an integer number of words N,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>where N = Ceil(B / 4) *</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When B is not a multiple of 4, a number of dummy bytes (4 N – B) shall be</td>
</tr>
<tr>
<td></td>
<td></td>
<td>padded after the last data byte.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Padded dummy bytes shall have the value 0.</td>
</tr>
<tr>
<td>N+2</td>
<td>CRC</td>
<td>32 bit CRC calculated over words 0 to (N+2)-1.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4x K29.7</td>
<td>End of packet indication.</td>
<td></td>
</tr>
</tbody>
</table>

* Ceil(x) returns the smallest integer ≥ x
Packet size: N + 6 words ((N * 4) + 24 bytes).

8.6.3 Acknowledgment Message Format

Acknowledgment messages are transmitted in the payload section of a data packet with packet type 0x03. The payload format is specified in the following table:
### Table 22 — Acknowledgment message format

<table>
<thead>
<tr>
<th>Word</th>
<th>Content</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4x K27.7</td>
<td>Start of packet indication (see section 8.4).</td>
<td></td>
</tr>
<tr>
<td>4x 0x03</td>
<td>Control acknowledge indication.</td>
<td></td>
</tr>
</tbody>
</table>
| 0        | 4x Code | Acknowledgment code (repeated 4 times):  
            **Success:**  
            0x00  Final, command executed OK, reply data is appended  
                   (i.e. acknowledgment of read command).  
            0x01  Final, command executed OK, No reply data is appended  
                   (i.e. acknowledgment of write command).  
            0x03  Final, control channel reset executed OK.  
            0x04  Wait. The time for the Host to wait shall be sent as a 4 byte integer in the reply data field, using the same packet structure as for a 0x00 acknowledgement. The value shall be in milliseconds, with a range of 100ms to 10s.  
            **Logical Errors (final acknowledgments):**  
            0x40  Invalid address.  
            0x41  Invalid data for the address.  
            0x42  Invalid control operation code.  
            0x43  Write attempted to a read-only address.  
            0x44  Read attempted from a write-only address.  
            0x45  Size field too large – command message (write) or acknowledgment message (read) would exceed packet size limit.  
            0x46  Incorrect size received, message size is inconsistent with message size indication.  
            0x47  Malformed packet.  
            **Physical Errors (final acknowledgments):**  
            0x80  Failed CRC test in last received command.  
            Others values are reserved for future use.  
            For all acknowledgment codes other than 0x00 or 0x04, the Length, Data and CRC fields shall be omitted and the end of packet indication shall be transmitted after the acknowledgment code. |
| 1        | Size    | Number of appended reply data bytes B, which shall be the same as the size field in the corresponding Control Command packet. B does not include any dummy bytes for padding. |
| 2 to (N-1)+2 | Data    | Reply data.  
            The size of this data section shall be an integer number of words N, where N = Ceil(B / 4) *  
            When B is not a multiple of 4, a number of dummy bytes (4 N – B) shall be padded after the last data byte.  
            Padded dummy bytes shall have the value 0. |
| N+2      | CRC     | 32 bit CRC calculated over data words 0 to (N+2)-1. |
| 4x K29.7 | End of packet indication. |

Packet size: \(N + 6\) words \(((N \times 4) + 24\) bytes).
8.6.4 Control Transaction Block Size

The total control packet size (i.e. from the first K27.7 to the last K29.7) shall not be more than the size set in bytes in ControlPacketSizeMax during Device discovery (see section 10.1.5).

8.7 Connection Test

The Host and Device shall provide connection test facilities to test the quality of the connection. Connection test in both directions is initiated and controlled by the Host.

8.7.1 Connection Test Methodology

Connection test makes use of test data packets between a Test Generator and a Test Receiver as depicted schematically in the following figure:

![Connection test methodology diagram]

The Test Generator shall transmit a Test Data Packet containing a known test pattern produced by a Sequence Generator, and increment the packet counter for each test packet transmitted. The Test Receiver shall compare the received test data packet content against its local Sequence Generator. The Test Receiver shall increment the Error Counter for each word that is different in the data packet, and increment the packet counter for each test packet received.

Comment: The test packet counters show how many test packets have been sent and received, so allowing a judgment to be made on the statistical meaning of the value in the error counter.

Comment: Both Device to Host and Host to Device connection tests can be run at the same time.

8.7.2 Connection Test Data Packet Payload Format

The Test Data Packet content shall consist of a continuous 4096 byte sequence generated by counting 16 times from 0 to 255 (4096 bytes, 1024 words) as illustrated in the following table:
Table 23 — Connection test packet format

<table>
<thead>
<tr>
<th>Word</th>
<th>Content</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>P0</td>
<td>K27.7</td>
<td>Start of packet indication (see section 8.4).</td>
</tr>
<tr>
<td>P1</td>
<td>K27.7</td>
<td></td>
</tr>
<tr>
<td>P2</td>
<td>K27.7</td>
<td></td>
</tr>
<tr>
<td>P3</td>
<td>K27.7</td>
<td></td>
</tr>
<tr>
<td>0x04</td>
<td>0x04</td>
<td>Connection test indication.</td>
</tr>
<tr>
<td>0</td>
<td>0x00</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0x04</td>
<td></td>
</tr>
<tr>
<td>0x05</td>
<td>0x06</td>
<td></td>
</tr>
<tr>
<td>0x07</td>
<td></td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>0xFC</td>
<td>Test data</td>
</tr>
<tr>
<td>0x01</td>
<td>0x02</td>
<td></td>
</tr>
<tr>
<td>0x03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1023</td>
<td>0xFC</td>
<td>End of packet indication.</td>
</tr>
<tr>
<td></td>
<td>0xFD</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0xFE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0xFF</td>
<td></td>
</tr>
</tbody>
</table>

Packet size: 1027 words (4108 bytes).

Comment: Using a PRBS (pseudo-random binary sequence) is of no use since 8B/10B line coding is applied to the generated data stream. For the purpose at hand a simple counter works equally well and greatly eases implementation.

The transmitter shall give priority to Control Data over Test Data for packet transmission.

Comment: This ensures that the control messages to start and stop connection test have priority over the test data.

8.7.3 Host to Device Connection Test

The Host shall implement a Test Generator and the Device a Test Receiver as shown in section 8.7.1. The Error Counter in the Device shall be implemented as bootstrap registers TestErrorCount[m] (where m is the connection number 0 .. n-1, or the value n for the optional high speed upconnection). The Host shall have read and write access to these registers allowing it to reset them. Similarly, the Device shall implement bootstrap registers TestPacketCountRx[m] to show the total packet count received during the test, which the Host shall have read and write access to.

The Host shall use a spacing of at least one IDLE word between each test data packet, and shall insert IDLE packets into the test sequence to meet the IDLE requirements (see section 8.2.5.1).

The Device shall process a received test data packet from the Host at all times regardless of the setting of the TestMode bootstrap register.

To test the Host to Device connections the Host shall execute the following operations:

- Reset the Device Error Counters and Packet Counters by writing the value 0 to TestErrorCount[m] and TestPacketCountRx[m] registers.
- Reset the Packet Counters at the Host transmitters that are involved in the test.
• Run the test for a pre-determined amount of time providing the desired coverage for each connection involved in the test.
• Read and process the Device TestErrorCount[m] and TestPacketCountRx[m] register values, and the Host Transmit Packet Counters, for each connection involved in the test.

8.7.4 Device to Host Connection Test

The Device shall implement a Test Generator and the Host a Test Receiver as shown in section 8.7.1. The Test Generator shall be controlled via the downconnection test control bootstrap register TestMode. The Host shall have read and write access to this register. When the register is set to 1 for Test Mode the Device shall transmit test data packets at a regular interval.

The spacing between test data packets shall be at least 16 word intervals allowing room for control data packets. The Device shall insert IDLE packets into the test sequence to meet the IDLE requirements (see section 8.2.5.1), but shall not transmit data other than connection test packets or control packets.

To test the Device to Host connections the Host shall execute the following operations:

• Stop the Device from transmitting streaming data if applicable (stop image acquisition).
• Reset the Device Packet Counters by writing the value 0 to TestPacketCountTx[m] registers (where m is the connection number 0 .. n-1, or the value n for the optional high speed upconnection).
• Reset the Error Counters and Packet Counters at the Host receivers that are involved in the test.
• Put the Device in Test Mode by writing 1 to the TestMode bootstrap register.
• Run the test for a pre-determined amount of time providing the desired coverage for each connection involved in the test.
• Read and process the Host Error Counter and Receive Packet Counter values, and the Device TestPacketCountTx[m] registers, for each connection involved in the test.
• Restore normal Device operation by writing 0 to the TestMode bootstrap register.
9 Data Streams

9.1 Overview
Stream channels shall be used to transfer data from Device to Host. Data shall be formed into streams before being split into packets to send over the physical connection(s), as described in section 8. Data stream formats are defined for rectangular and non-rectangular images, but other types may be defined in future, such as for auxiliary data, compressed images and audio.
If a Device generates multiple images (e.g. multiple ROIs), each image is formed into a separate stream (see section 9.4.3).
Similarly, data from multiple taps are formed into separate streams (see section 9.4.5).

9.2 Stream Format
A stream comprises a header giving the stream type and information about the following data, then the data itself.
A stream uses K-code K28.3 as a stream marker, in addition to the K-codes used for packet transfer (see section 8.2.1). This K-code is used to identify the stream header and important points in the data stream, such as start-of-line (for an image stream).
Comment: The format of a stream is designed to give a low overhead in terms of the number of additional characters needed for the stream header and stream markers, so allowing very high speed transfer of very small image sizes.

9.3 Stream ID
Each stream from a Device shall have a unique stream ID. The Device shall have a fixed set of streams it can produce, each with an allocated static stream ID.
Comment: The GenICam XML file and product documentation give the stream information for the Device.
Stream IDs range from 0 to 255.
It is recommended that the primary stream from a Device has stream ID = 0.


\section*{9.4 Image Streams}

Image streams are formed from lines. An image header is used to inform the Host about the format of succeeding image lines. A line marker separates data from successive lines.

A typical image stream is illustrated in the following figure:

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{image_stream_structure.png}
\caption{Image stream structure}
\end{figure}

\subsection*{9.4.1 Pixel Types}

This specification is based on the pixel format names and definition of the Pixel Format Naming Convention (PFNC) (section 2.2 ref 7), which is part of the GenICam standard.

The Device shall use pixel formats defined in Table 25, and its XML file shall list the formats it supports using the names given in the “PFNC Name” column. Bit level coding of the pixel formats and packing over the CoaXPress link is CoaXPress specific (i.e. not as defined by the PFNC), as are some Planar formats which are not defined in the PFNC.

\textit{Comment:} The CoaXPress v1.1 standard required the use of the suffix “pmsb” on all the packed format names, to comply with the GenICam PFNC and SFNC. This requirement was ignored by most vendors, and the PFNC has now been revised to remove this requirement. CoaXPress v1.1.1 reflects this change to the PFNC.

The Host can map the Device’s data to any pixel format in the PFNC, and the resulting image generated by the Host shall be precisely as defined by the PFNC.

Both Devices and Hosts shall support at least one of the types listed in Table 25 over the CoaXPress link.

\textit{Comment:} A typical frame grabber Host would support many types.

\textit{Comment:} YUV and YCbCr are clearly defined to avoid the usual uncertainty with these formats.

\subsection*{9.4.1.1 Pixel Formats}

The pixel format code is formed as shown in the following table. This code is used in the \textit{PixelF} field in the image headers. Note that the value 0x0000 is reserved for “raw” data that does not match any defined format, such as user-specific formats.
Table 24 — PixelF coding

<table>
<thead>
<tr>
<th>Bits</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>15..8</td>
<td>Data Type</td>
</tr>
<tr>
<td>7..4</td>
<td>Sub-type</td>
</tr>
<tr>
<td>3..0</td>
<td>Data Width</td>
</tr>
</tbody>
</table>

Comment: Many format codes are spare to allow future formats and data widths to be added.

Similarly, the format name is concatenated from the Data Type, Sub-type, and data width. e.g. BayerGR8 is Bayer data, with order GR, and 8 bits per component.

The data types and widths are defined in the following section, and the resulting PixelF values for most formats (only a selection of planar formats are listed for clarity) are shown in the following table, along with the PFNC name as used by GenICam.

Table 25 — PixelF values

<table>
<thead>
<tr>
<th>Pixel format</th>
<th>PFNC Name</th>
<th>PixelF Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw</td>
<td>Raw</td>
<td>0x0000</td>
</tr>
<tr>
<td>Mono8</td>
<td>Mono8</td>
<td>0x0101</td>
</tr>
<tr>
<td>Mono10</td>
<td>Mono10</td>
<td>0x0102</td>
</tr>
<tr>
<td>Mono12</td>
<td>Mono12</td>
<td>0x0103</td>
</tr>
<tr>
<td>Mono14</td>
<td>Mono14</td>
<td>0x0104</td>
</tr>
<tr>
<td>Mono16</td>
<td>Mono16</td>
<td>0x0105</td>
</tr>
<tr>
<td>Planar1_8</td>
<td>-</td>
<td>0x0211</td>
</tr>
<tr>
<td>Planar1_10</td>
<td>-</td>
<td>0x0212</td>
</tr>
<tr>
<td>Planar1_12</td>
<td>-</td>
<td>0x0213</td>
</tr>
<tr>
<td>Planar1_14</td>
<td>-</td>
<td>0x0214</td>
</tr>
<tr>
<td>Planar1_16</td>
<td>-</td>
<td>0x0215</td>
</tr>
<tr>
<td>Planar2_8</td>
<td>-</td>
<td>0x0221</td>
</tr>
<tr>
<td>Planar2_10</td>
<td>-</td>
<td>0x0222</td>
</tr>
<tr>
<td>Planar2_12</td>
<td>-</td>
<td>0x0223</td>
</tr>
<tr>
<td>Planar2_14</td>
<td>-</td>
<td>0x0224</td>
</tr>
<tr>
<td>Planar2_16</td>
<td>-</td>
<td>0x0225</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>Planar15_8</td>
<td>-</td>
<td>0x02F1</td>
</tr>
<tr>
<td>Planar15_10</td>
<td>-</td>
<td>0x02F2</td>
</tr>
<tr>
<td>Planar15_12</td>
<td>-</td>
<td>0x02F3</td>
</tr>
<tr>
<td>Planar15_14</td>
<td>-</td>
<td>0x02F4</td>
</tr>
<tr>
<td>Planar15_16</td>
<td>-</td>
<td>0x02F5</td>
</tr>
</tbody>
</table>

*continued on next page*
PixelF values continued…

<table>
<thead>
<tr>
<th>Pixel format</th>
<th>PFNC Name</th>
<th>PixelF Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>BayerGR8</td>
<td>BayerGR8</td>
<td>0x0311</td>
</tr>
<tr>
<td>BayerGR10</td>
<td>BayerGR10</td>
<td>0x0312</td>
</tr>
<tr>
<td>BayerGR12</td>
<td>BayerGR12</td>
<td>0x0313</td>
</tr>
<tr>
<td>BayerGR14</td>
<td>BayerGR14</td>
<td>0x0314</td>
</tr>
<tr>
<td>BayerGR16</td>
<td>BayerGR16</td>
<td>0x0315</td>
</tr>
<tr>
<td>BayerRG8</td>
<td>BayerRG8</td>
<td>0x0321</td>
</tr>
<tr>
<td>BayerRG10</td>
<td>BayerRG10</td>
<td>0x0322</td>
</tr>
<tr>
<td>BayerRG12</td>
<td>BayerRG12</td>
<td>0x0323</td>
</tr>
<tr>
<td>BayerRG14</td>
<td>BayerRG14</td>
<td>0x0324</td>
</tr>
<tr>
<td>BayerRG16</td>
<td>BayerRG16</td>
<td>0x0325</td>
</tr>
<tr>
<td>BayerGB8</td>
<td>BayerGB8</td>
<td>0x0331</td>
</tr>
<tr>
<td>BayerGB10</td>
<td>BayerGB10</td>
<td>0x0332</td>
</tr>
<tr>
<td>BayerGB12</td>
<td>BayerGB12</td>
<td>0x0333</td>
</tr>
<tr>
<td>BayerGB14</td>
<td>BayerGB14</td>
<td>0x0334</td>
</tr>
<tr>
<td>BayerGB16</td>
<td>BayerGB16</td>
<td>0x0335</td>
</tr>
<tr>
<td>BayerBG8</td>
<td>BayerBG8</td>
<td>0x0341</td>
</tr>
<tr>
<td>BayerBG10</td>
<td>BayerBG10</td>
<td>0x0342</td>
</tr>
<tr>
<td>BayerBG12</td>
<td>BayerBG12</td>
<td>0x0343</td>
</tr>
<tr>
<td>BayerBG14</td>
<td>BayerBG14</td>
<td>0x0344</td>
</tr>
<tr>
<td>BayerBG16</td>
<td>BayerBG16</td>
<td>0x0345</td>
</tr>
<tr>
<td>RGB8</td>
<td>RGB8</td>
<td>0x0401</td>
</tr>
<tr>
<td>RGB10</td>
<td>RGB10</td>
<td>0x0402</td>
</tr>
<tr>
<td>RGB12</td>
<td>RGB12</td>
<td>0x0403</td>
</tr>
<tr>
<td>RGB14</td>
<td>RGB14</td>
<td>0x0404</td>
</tr>
<tr>
<td>RGB16</td>
<td>RGB16</td>
<td>0x0405</td>
</tr>
<tr>
<td>RGBA8</td>
<td>RGBa8</td>
<td>0x0501</td>
</tr>
<tr>
<td>RGBA10</td>
<td>RGBa10</td>
<td>0x0502</td>
</tr>
<tr>
<td>RGBA12</td>
<td>RGBa12</td>
<td>0x0503</td>
</tr>
<tr>
<td>RGBA14</td>
<td>RGBa14</td>
<td>0x0504</td>
</tr>
<tr>
<td>RGBA16</td>
<td>RGBa16</td>
<td>0x0505</td>
</tr>
<tr>
<td>YUV411_8</td>
<td>YUV411_8</td>
<td>0x0611</td>
</tr>
<tr>
<td>YUV411_10</td>
<td>YUV411_10</td>
<td>0x0612</td>
</tr>
<tr>
<td>YUV411_12</td>
<td>YUV411_12</td>
<td>0x0613</td>
</tr>
<tr>
<td>YUV411_14</td>
<td>YUV411_14</td>
<td>0x0614</td>
</tr>
<tr>
<td>YUV411_16</td>
<td>YUV411_16</td>
<td>0x0615</td>
</tr>
<tr>
<td>YUV422_8</td>
<td>YUV422_8</td>
<td>0x0621</td>
</tr>
<tr>
<td>YUV422_10</td>
<td>YUV422_10</td>
<td>0x0622</td>
</tr>
<tr>
<td>YUV422_12</td>
<td>YUV422_12</td>
<td>0x0623</td>
</tr>
</tbody>
</table>

continued on next page …
PixelF values continued…

<table>
<thead>
<tr>
<th>Pixel format</th>
<th>PFNC Name</th>
<th>PixelF Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>YUV422_14</td>
<td>YUV422_14</td>
<td>0x0624</td>
</tr>
<tr>
<td>YUV422_16</td>
<td>YUV422_16</td>
<td>0x0625</td>
</tr>
<tr>
<td>YUV444_8</td>
<td>YUV8</td>
<td>0x0631</td>
</tr>
<tr>
<td>YUV444_10</td>
<td>YUV10</td>
<td>0x0632</td>
</tr>
<tr>
<td>YUV444_12</td>
<td>YUV12</td>
<td>0x0633</td>
</tr>
<tr>
<td>YUV444_14</td>
<td>YUV14</td>
<td>0x0634</td>
</tr>
<tr>
<td>YUV444_16</td>
<td>YUV16</td>
<td>0x0635</td>
</tr>
<tr>
<td>YCbCr601_411_8</td>
<td>YCbCr601_411_8</td>
<td>0x0711</td>
</tr>
<tr>
<td>YCbCr601_411_10</td>
<td>YCbCr601_411_10</td>
<td>0x0712</td>
</tr>
<tr>
<td>YCbCr601_411_12</td>
<td>YCbCr601_411_12</td>
<td>0x0713</td>
</tr>
<tr>
<td>YCbCr601_411_14</td>
<td>YCbCr601_411_14</td>
<td>0x0714</td>
</tr>
<tr>
<td>YCbCr601_411_16</td>
<td>YCbCr601_411_16</td>
<td>0x0715</td>
</tr>
<tr>
<td>YCbCr601_422_8</td>
<td>YCbCr601_422_8</td>
<td>0x0721</td>
</tr>
<tr>
<td>YCbCr601_422_10</td>
<td>YCbCr601_422_10</td>
<td>0x0722</td>
</tr>
<tr>
<td>YCbCr601_422_12</td>
<td>YCbCr601_422_12</td>
<td>0x0723</td>
</tr>
<tr>
<td>YCbCr601_422_14</td>
<td>YCbCr601_422_14</td>
<td>0x0724</td>
</tr>
<tr>
<td>YCbCr601_422_16</td>
<td>YCbCr601_422_16</td>
<td>0x0725</td>
</tr>
<tr>
<td>YCbCr601_444_8</td>
<td>YCbCr601_8</td>
<td>0x0731</td>
</tr>
<tr>
<td>YCbCr601_444_10</td>
<td>YCbCr601_10</td>
<td>0x0732</td>
</tr>
<tr>
<td>YCbCr601_444_12</td>
<td>YCbCr601_12</td>
<td>0x0733</td>
</tr>
<tr>
<td>YCbCr601_444_14</td>
<td>YCbCr601_14</td>
<td>0x0734</td>
</tr>
<tr>
<td>YCbCr601_444_16</td>
<td>YCbCr601_16</td>
<td>0x0735</td>
</tr>
<tr>
<td>YCbCr709_411_8</td>
<td>YCbCr709_411_8</td>
<td>0x0811</td>
</tr>
<tr>
<td>YCbCr709_411_10</td>
<td>YCbCr709_411_10</td>
<td>0x0812</td>
</tr>
<tr>
<td>YCbCr709_411_12</td>
<td>YCbCr709_411_12</td>
<td>0x0813</td>
</tr>
<tr>
<td>YCbCr709_411_14</td>
<td>YCbCr709_411_14</td>
<td>0x0814</td>
</tr>
<tr>
<td>YCbCr709_411_16</td>
<td>YCbCr709_411_16</td>
<td>0x0815</td>
</tr>
<tr>
<td>YCbCr709_422_8</td>
<td>YCbCr709_422_8</td>
<td>0x0821</td>
</tr>
<tr>
<td>YCbCr709_422_10</td>
<td>YCbCr709_422_10</td>
<td>0x0822</td>
</tr>
<tr>
<td>YCbCr709_422_12</td>
<td>YCbCr709_422_12</td>
<td>0x0823</td>
</tr>
<tr>
<td>YCbCr709_422_14</td>
<td>YCbCr709_422_14</td>
<td>0x0824</td>
</tr>
<tr>
<td>YCbCr709_422_16</td>
<td>YCbCr709_422_16</td>
<td>0x0825</td>
</tr>
<tr>
<td>YCbCr709_444_8</td>
<td>YCbCr709_8</td>
<td>0x0831</td>
</tr>
<tr>
<td>YCbCr709_444_10</td>
<td>YCbCr709_10</td>
<td>0x0832</td>
</tr>
<tr>
<td>YCbCr709_444_12</td>
<td>YCbCr709_12</td>
<td>0x0833</td>
</tr>
<tr>
<td>YCbCr709_444_14</td>
<td>YCbCr709_14</td>
<td>0x0834</td>
</tr>
<tr>
<td>YCbCr709_444_16</td>
<td>YCbCr709_16</td>
<td>0x0835</td>
</tr>
</tbody>
</table>
9.4.1.2 Data Width / Packing

This is the width in bits of each pixel, or for color formats the width in bits of each color component of the pixel. The width also defines the packing mode for the data. This is defined in the following table:

<table>
<thead>
<tr>
<th>Data Width</th>
<th>Bit Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 bit</td>
<td>0001</td>
</tr>
<tr>
<td>10 bit</td>
<td>0010</td>
</tr>
<tr>
<td>12 bit</td>
<td>0011</td>
</tr>
<tr>
<td>14 bit</td>
<td>0100</td>
</tr>
<tr>
<td>16 bit</td>
<td>0101</td>
</tr>
</tbody>
</table>

9.4.1.3 Mono Format

This is used for luminance data. This has no sub-types. This is defined in the following table:

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Bit Definition</th>
<th>Sub-type</th>
<th>Bit Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mono</td>
<td>00000001</td>
<td>-</td>
<td>0000</td>
</tr>
</tbody>
</table>

9.4.1.4 Planar Format

This is used for planar data, such as individual red, green or blue planes, additional alpha (overlay) planes, or the separate planes in YUV420. The use of each plane by the Device is implementation dependent and beyond the scope of this specification, but planar RGB and YUV420 images shall follow the standard usage given. This is defined in the following table:

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Bit Definition</th>
<th>Sub-type</th>
<th>Bit Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planar</td>
<td>00000010</td>
<td>Plane 1 Standard usage: R, Y</td>
<td>0001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Plane 2 Standard usage: G, U, Cb</td>
<td>0010</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Plane 3 Standard usage: B, V, Cr</td>
<td>0011</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Plane 4</td>
<td>0100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Plane 15</td>
<td>1111</td>
</tr>
</tbody>
</table>
9.4.1.5 Bayer Format

This is used for Bayer data. This is defined in the following table:

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Bit Definition</th>
<th>Sub-type</th>
<th>Bit Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bayer</td>
<td>00000011</td>
<td>GR</td>
<td>0001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1st line transmission order G, R</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2nd line transmission order B, G</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RG</td>
<td>0010</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1st line transmission order R, G</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2nd line transmission order G, B</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>GB</td>
<td>0011</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1st line transmission order G, B</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2nd line transmission order R, G</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>BG</td>
<td>0100</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1st line transmission order B, G</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2nd line transmission order G, R</td>
<td></td>
</tr>
</tbody>
</table>

9.4.1.6 RGB Format

This is used for RGB data, transmitted in the order red, green, blue. This has no sub-types. This is defined in the following table:

Comment: No other component orders (e.g. BGR) are supported by CoaXPress – they can be generated in the Host if required. A commonly used format like RGB24 is simply this RGB format with 8 bit data per color.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Bit Definition</th>
<th>Sub-type</th>
<th>Bit Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGB</td>
<td>00000100</td>
<td>-</td>
<td>0000</td>
</tr>
</tbody>
</table>

9.4.1.7 RGBA Format

This is used for RGBA data, where “A” is the alpha (or overlay) plane, transmitted in the order red, green, blue, alpha. This has no sub-types. This is defined in the following table:

Comment: No other component orders (e.g. ABGR) are supported by CoaXPress.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Bit Definition</th>
<th>Sub-type</th>
<th>Bit Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>RGBA</td>
<td>00000101</td>
<td>-</td>
<td>0000</td>
</tr>
</tbody>
</table>
9.4.1.8 YUV Format

This is used for YUV data, which for purposes of this standard has the full range of $0..2^{n-1}$, rather than being limited (e.g. $0..255$ rather than $16..235$), and shall be as defined by the formulae in the comment below. This is defined in the following table:

Table 32 — YUV definition

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Bit Definition</th>
<th>Sub-type</th>
<th>Bit Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>YUV</td>
<td>00000110</td>
<td>411</td>
<td>0001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transmission order Y, Y, U, Y, V</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>422</td>
<td>0010</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transmission order Y, U, Y, V</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>444</td>
<td>0011</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transmission order Y, U, V</td>
<td></td>
</tr>
</tbody>
</table>

Comment: Strictly there is not a digital YUV standard, but the following formulae are often used to map RGB data with a range $0..255$ to YUV data. In the resulting 8 bit data, $Y$ has the range $0..255$; $U$ and $V$ are signed with a range $16..240$ and 128 representing zero.

\[
Y = 0.299 R + 0.587 G + 0.114 B \\
U = -0.169 R - 0.331 G + 0.499 B + 128 \\
V = 0.499 R - 0.418 G - 0.0813 B + 128
\]

9.4.1.9 YCbCr601 Format

This is used for YCbCr data, as specified by ITU-R BT.601 (section 2.2 ref 4) for 8 and 10 bit data. This is defined in the following table:

Table 33 — YCbCr601 definition

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Bit Definition</th>
<th>Sub-type</th>
<th>Bit Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>YCbCr601</td>
<td>00000111</td>
<td>411</td>
<td>0001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transmission order Y, Y, Cb, Y, Y, Cr</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>422</td>
<td>0010</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transmission order Y, Cb, Y, Cr</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>444</td>
<td>0011</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transmission order Y, Cb, Cr</td>
<td></td>
</tr>
</tbody>
</table>

Comment: This is the standard format for standard definition broadcast video. For 8 bit data, $Y$ has the range $16..235$; $Cb$ and $Cr$ are signed with a range $16..240$ and 128 representing zero. The ITU-R BT.601 formulae can be simplified to the following for gamma corrected RGB data with a range $16..240$:

\[
Y = 0.299 R + 0.587 G + 0.114 B \\
Cb = -0.173 R - 0.339 G + 0.511 B + 128 \\
Cr = 0.511 R - 0.428 G - 0.0832 B + 128
\]
9.4.1.10  YCbCr709 Format

This is used for YCbCr data, as specified by ITU-R BT.709 (section 2.2 ref 5). This is defined in the following table:

Table 34 — YCbCr709 definition

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Bit Definition</th>
<th>Sub-type</th>
<th>Bit Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>YCbCr709</td>
<td>00001000</td>
<td>411 Transmission order Y, Y, Cb, Y, Y, Cr</td>
<td>0001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>422 Transmission order Y, Cb, Y, Cr</td>
<td>0010</td>
</tr>
<tr>
<td></td>
<td></td>
<td>444 Transmission order Y, Cb, Cr</td>
<td>0011</td>
</tr>
</tbody>
</table>

Comment: This is the standard format for high definition broadcast video. For 8 bit data, Y has the range 16..235; Cb and Cr are signed with a range 16..240 and 128 representing zero. The ITU-R BT.709 formulae can be simplified to the following for gamma corrected RGB data with a range 16..240:

\[
Y = 0.213R + 0.715G + 0.0722B \\
Cb = -0.117R -0.394G + 0.511B + 128 \\
Cr = 0.511R -0.465G -0.0469B + 128
\]
9.4.2 Pixel Packing

The link protocol enforces maximum density packing in order to maximize throughput for the available bandwidth. Packing schemes are defined in following tables.

Pixels shall be packed within a line. The first pixel of a new line shall be stored into P0.

For some packing schemes the packed data size may not be an integer number of words across a line. In such a case the number of data words shall be rounded up to the nearest integer value, and the unused bits of the last data word shall be set 0. Note that both the number of words per line and pixels per line are provided by the preceding header or line marker.

The following figures show packing definitions for 8, 10, 12, 14 and 16 bit pixels (in the case of mono or planar formats) or components (in the case of multi-component formats such as RGB):

![Figure 27 — Packing of 8 bit pixels or components](image)

![Figure 28 — Packing of 10 bit pixels or components](image)
Figure 29 — Packing of 12 bit pixels or components

<table>
<thead>
<tr>
<th>P0</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>11</td>
<td>D(0)</td>
</tr>
<tr>
<td>0</td>
<td>11</td>
<td>D(1)</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>11</td>
<td>D(2)</td>
</tr>
<tr>
<td>D(2)</td>
<td>D(3)</td>
<td>D(4)</td>
<td>D(5)</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>D(5)</td>
<td>D(6)</td>
<td>D(7)</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 30 — Packing of 14 bit pixels or components

<table>
<thead>
<tr>
<th>P0</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>15</td>
<td>D(0)</td>
</tr>
<tr>
<td>D(0)</td>
<td>D(1)</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 31 — Packing of 16 bit pixels or components

Comment: These packing modes define the data transmitted over CoaXPress. A frame grabber in the Host may process this data and provide different formats to the Host memory and/or display.
Pixel sizes that are in-between the defined packing sizes shall be MSB aligned into the next larger defined size. The unused least significant bits of the resulting pixel can either be filled with zeros or a dither pattern (implementation dependent). An example of 15 bit data being fitted into 16 bits prior to packing is shown in the following figure:

Figure 32 — Packing of 15 into 16 bits
Comment: Again this is for the data transmitted over CoaXPress, and simplifies hardware handling of varying width pixel data.

9.4.3 Multiple Images

A Device that generates more than one image shall form each image into a separate stream.

Comment: This allows for Devices that output several small regions of interest (ROI), and Devices that have more than one sensor (e.g. visible and IR).

9.4.4 Image Scan Format

Horizontal image scanning shall be fixed: Left to right.

Vertical image scanning can be intermixed using the concept of taps.

Comment: Devices that use multi-tap image sensors having opposite horizontal scan directions can relatively easy perform pixel rearrangement to obtain a horizontal single scan direction output. Using intermixed horizontal scan directions leads to complex definitions of how an image stream must be decoded at the Host, especially when combined with dynamic variable ROI capabilities. This is avoided by demanding horizontal rearrangement to be performed by the Device. Vertical multi tap scanning rearrangement on the other hand is better done at the Host giving an overall lower latency delay. Furthermore it prevents the need for frame size buffers at the Device to do the vertical rearrangement.
9.4.5 Tap Concept

Taps apply to vertical scanning only (see comment above). The number of taps shall be read or set by the Host via a Device register. The tap geometry naming follows the GenICam SFNC Tap Geometry naming, as shown in the following table:

<table>
<thead>
<tr>
<th>Name</th>
<th>Valid Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZoneY</td>
<td>1, 2</td>
<td>The number of zones across the vertical direction.</td>
</tr>
<tr>
<td>TapY</td>
<td>Null, 2</td>
<td>The number of consecutive lines in the vertical direction that output pixels simultaneously from a zone. ‘Null’ means the default of a single line.</td>
</tr>
<tr>
<td>ExtractorY</td>
<td>Null, E</td>
<td>The location of the pixel extractors across vertical direction. ‘Null’ means the default of pixel extractors all at the top line. ‘E’ means pixel extractors are at both top and bottom lines.</td>
</tr>
</tbody>
</table>

The tap format is defined: 1X-<ZoneY>Y<TapY><ExtractorY>

The following tap scan geometries are supported:

- 1X-1Y (default single tap format, also used for line scan)
- 1X-1Y2
- 1X-2YE

Both Devices and Hosts shall support at least one of these tap formats.

Comment: A typical frame grabber Host would support all tap formats.

Scanning examples for both supported multi-tap geometries are illustrated in the following figure:

Each tap shall form a separate stream.
The tap format code is formed as shown in the following table. This code is used in the TapG field in the image headers.

### Table 36 — TapG coding

<table>
<thead>
<tr>
<th>Bits</th>
<th>Usage</th>
<th>Bit Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>15..12</td>
<td>ThisTap</td>
<td>The tap number this stream is from. “N-1” coding, i.e.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0000: Tap 1. 0001: Tap 2. Other values are reserved.</td>
</tr>
<tr>
<td>11..10</td>
<td>Reserved</td>
<td>00</td>
</tr>
<tr>
<td>9..6</td>
<td>ZoneY</td>
<td>“N-1” coding, i.e. 0000: 1. 0001: 2. Other values are reserved.</td>
</tr>
<tr>
<td>5..2</td>
<td>TapY</td>
<td>“N-1” coding, i.e. 0000: Null. 0001: 2. Other values are reserved.</td>
</tr>
<tr>
<td>1..0</td>
<td>ExtractorY</td>
<td>00: Null. 01: E. Other values are reserved.</td>
</tr>
</tbody>
</table>

The resulting TapG values for the defined formats are shown in the following table:

### Table 37 — TapG values

<table>
<thead>
<tr>
<th>Tap format</th>
<th>TapG Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1X-1Y</td>
<td>0x0000</td>
</tr>
<tr>
<td>1X-1Y2, tap 1</td>
<td>0x0004</td>
</tr>
<tr>
<td>1X-1Y2, tap 2</td>
<td>0x1004</td>
</tr>
<tr>
<td>1X-2YE, tap 1</td>
<td>0x0041</td>
</tr>
<tr>
<td>1X-2YE, tap 2</td>
<td>0x1041</td>
</tr>
</tbody>
</table>
9.4.6 Rectangular Image Stream

This section applies to standard uncompressed rectangular area scan or line scan images. See section 9.4.7 for arbitrary shaped image streams.

9.4.6.1 Transmission Rules

A rectangular image is defined as one where:

- The size of the image (pixels per line, lines per frame) is known by the Device before it sends the image, so that it can generate a header.
- Each line in the image is the same length.
- Each line in the image has the same horizontal offset with respect to the left hand pixel of the full Device image.
- Each pixel in the image is the same type (e.g. color, mono etc).
- The image can be progressive scan or interlaced.
- The image can be Area Scan or Line Scan.

Comment: Successive images in the stream can be different sizes, because they have their own image header.

The following rules shall be used to form a rectangular image stream from an area scan Device:

- A rectangular image header shall be sent before the first line of each area scan image.
  
  Comment: The header therefore acts as a frame marker. The header contains critical information needed by the Host to decode succeeding image line data. The contents of the header are therefore redundantly encoded to guarantee proper decoding in case of a single bit error during transmission.

- A rectangular image line marker shall be sent before each line.
- Image data shall not be packed across line boundaries. Therefore the first pixel of a new line shall be stored into P0. See section 9.4.2.

The following rules shall be used to form a rectangular image stream from a line scan Device:

- A rectangular image header shall be sent before the first line from a line scan Device. Subsequent lines can be preceded by a rectangular image line marker or a rectangular image header.
  
  Comment: The line marker reduces overhead, particularly with short lines.

- Additionally, a line scan Device shall send a rectangular image header at least every 200ms, or once per line if the time between lines is more than 200ms.
  
  Comment: This allows recovery in the event that serious multiple bit errors corrupt the initial header.

- Image data shall not be packed across line boundaries. Therefore the first pixel of a new line shall be stored into P0. See section 9.4.2.
9.4.6.2 Rectangular Image Header

The header contains critical information needed by the Host to decode succeeding image line data. The contents of the header are therefore redundantly encoded to allow proper decoding in case of a single bit error during transmission.

The image header shall be as shown in the following table:

<table>
<thead>
<tr>
<th>Word</th>
<th>Content</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4x K28.3</td>
<td>Stream Marker.</td>
</tr>
<tr>
<td>2</td>
<td>4x 0x01</td>
<td>Rectangular Image Header indication.</td>
</tr>
<tr>
<td>3</td>
<td>4x StreamID</td>
<td>Unique stream ID.</td>
</tr>
<tr>
<td>4</td>
<td>4x SourceTag(15:8)</td>
<td>16 bit source image index. Incremented for each transferred image, wraparound to 0 at 0xFFFF. The same number shall be used by each stream containing data relating to the same image.</td>
</tr>
<tr>
<td>5</td>
<td>4x SourceTag (7:0)</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>4x Xsize(23:16)</td>
<td>24 bit value representing the image width in pixels.</td>
</tr>
<tr>
<td>7</td>
<td>4x Xsize(15:8)</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>4x Xsize(7:0)</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>4x Xoffs(23:16)</td>
<td>24 bit value representing the horizontal offset in pixels of the image with respect to the left hand pixel of the full Device image.</td>
</tr>
<tr>
<td>10</td>
<td>4x Xoffs(15:8)</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>4x Xoffs(7:0)</td>
<td>Comment: Xoffs and Yoffs tell the Host where this image is with respect to the full Device image size.</td>
</tr>
<tr>
<td>12</td>
<td>4x Ysize(23:16)</td>
<td>24 bit value representing the image height in pixels. This value shall be set to 0 for line scan images.</td>
</tr>
<tr>
<td>13</td>
<td>4x Ysize(15:8)</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>4x Ysize(7:0)</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>4x Yoffs(23:16)</td>
<td>24 bit value representing the vertical offset in pixels of the image with respect to the top line of the full Device image. This value shall be set to 0 for line scan images.</td>
</tr>
<tr>
<td>16</td>
<td>4x Yoffs(15:8)</td>
<td>Comment: Xoffs and Yoffs tell the Host where this image is with respect to the full Device image size. As it represents the full Device image, it is the same for both fields of an interlaced image.</td>
</tr>
<tr>
<td>17</td>
<td>4x Yoffs(7:0)</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>4x DsizeL(23:16)</td>
<td>24 bit value representing the number of data words per image line.</td>
</tr>
<tr>
<td>19</td>
<td>4x DsizeL(15:8)</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>4x DsizeL(7:0)</td>
<td></td>
</tr>
</tbody>
</table>

continued on next page …
<table>
<thead>
<tr>
<th>Word</th>
<th>Content</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>4x PixelF(15:8)</td>
<td>16 bit value representing the pixel format (see section 9.4.1.1).</td>
</tr>
<tr>
<td>22</td>
<td>4x PixelF(7:0)</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>4x TapG(15:8)</td>
<td>16 bit value representing the tap geometry (see section 9.4.5).</td>
</tr>
<tr>
<td>24</td>
<td>4x TapG(7:0)</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>4x Flags</td>
<td>Image flags: (8 bits, repeated 4 times)</td>
</tr>
<tr>
<td></td>
<td>Bits: Usage:</td>
<td></td>
</tr>
</tbody>
</table>
|      | 1:0 Interlacing: | 0x0 None (Progressive scan or line scan)  
|      |               | 0x1 Interlaced, first line after header from field 1 *                     |
|      |               | 0x2 Interlaced, first line after header from field 2                        |
|      |               | 0x3 Reserved for future use                                                  |
|      | 7:2 Reserved for future use, set to 0                                       |

* Field 1 is defined as the field that includes the top image line.

---

**Figure 34 — Example 8x8 ROI in a VGA sized sensor**
9.4.6.3 Rectangular Image Line Marker

The line marker shall be as shown in the following table:

Table 39 — Rectangular image line marker

<table>
<thead>
<tr>
<th>Word</th>
<th>Content</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4x K28.3</td>
<td>Stream Marker.</td>
</tr>
<tr>
<td>2</td>
<td>4x 0x02</td>
<td>Rectangular line marker.</td>
</tr>
</tbody>
</table>
9.4.7  Arbitrary Image Stream

This section applies to arbitrary shaped area scan uncompressed images. See section 9.4.6 for standard rectangular image streams. Compared to a rectangular image the X offset, X size and DsizeL parameters are moved from the header to the line marker.

Comment: This stream format could also be used for rectangular images. This is not recommended because there is a significant additional per-line overhead, which would reduce the efficiency particularly for small images.

9.4.7.1  Transmission Rules

An arbitrary shaped image is defined as one where:

- The number of lines per frame is known by the Device before it sends the image, so that it can generate a header.
- The number of pixels per line is known by the Device before it sends the line, so that it can generate a line marker.
- Each line in the image can be a different length.
- Each line in the image can have a different horizontal offset with respect to the left hand pixel of the full Device image.
- Each pixel in the image is the same type (e.g. color, mono etc).
- The image can be progressive scan or interlaced.
- The image can be Area Scan or Line Scan.

Comment: Successive images in the stream can have different numbers of lines per frame, because they have their own image header.

The following rules shall be used to form an arbitrary image stream from an area scan Device:

- An arbitrary image header shall be sent before the first line of each area scan image.
  Comment: The header therefore acts as a frame marker. The header contains critical information needed by the Host to decode succeeding image line data. The contents of the header are therefore redundantly encoded to guarantee proper decoding in case of a single bit error during transmission.

- An arbitrary line marker shall be sent before each line.
- Image data shall not be packed across line boundaries. Therefore the first pixel of a new line shall be stored into P0. See section 9.4.2.

The following rules shall be used to form an arbitrary image stream from a line scan Device:

- An arbitrary image header shall be sent before the first line from a line scan Device. Subsequent lines can be preceded by an arbitrary line marker or an arbitrary image header.
  Comment: The line marker reduces overhead, particularly with short lines.

- Additionally, a line scan Device shall send an arbitrary image header at least every 200ms, or once per line if the time between lines is more than 200ms.
  Comment: This allows recovery in the event that serious multiple bit errors corrupt the initial header.
- Image data shall not be packed across line boundaries. Therefore the first pixel of a new line shall be stored into P0. See section 9.4.2.

### 9.4.7.2 Arbitrary Image Header

The header contains critical information needed by the Host to decode succeeding image line data. The contents of the header are therefore redundantly encoded to guarantee proper decoding in case of a single bit error during transmission.

The image header shall be as shown in the following table:

<table>
<thead>
<tr>
<th>Word</th>
<th>Content</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4x K28.3</td>
<td>Stream Marker.</td>
</tr>
<tr>
<td>2</td>
<td>4x 0x03</td>
<td>Arbitrary Image Header indication.</td>
</tr>
<tr>
<td>3</td>
<td>4x StreamID</td>
<td>Unique stream ID.</td>
</tr>
<tr>
<td>4</td>
<td>4x SourceTag (15:8)</td>
<td>16 bit source image index. Incremented for each transferred image, wraparound to 0 at 0xFFFF. The same number shall be used by each stream containing data relating to the same image.</td>
</tr>
<tr>
<td>5</td>
<td>4x SourceTag (7:0)</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>4x Ysize(23:16)</td>
<td>24 bit value representing the image height in pixels.</td>
</tr>
<tr>
<td>7</td>
<td>4x Ysize(15:8)</td>
<td>This value shall be set to 0 for line scan images.</td>
</tr>
<tr>
<td>8</td>
<td>4x Ysize(7:0)</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>4x Yoffs(23:16)</td>
<td>24 bit value representing the vertical offset in pixels of the image with respect to the top line of the full Device image.</td>
</tr>
<tr>
<td>10</td>
<td>4x Yoffs(15:8)</td>
<td>This value shall be set to 0 for line scan images. <strong>Comment:</strong> Yoffs (in the line marker) and Yoffs tell the Host where this image is with respect to the full Device image size. As it represents the full Device image, it is the same for both fields of an interlaced image.</td>
</tr>
<tr>
<td>11</td>
<td>4x Yoffs(7:0)</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>4x PixelF(15:8)</td>
<td>16 bit value representing the pixel format (see section 9.4.1.1).</td>
</tr>
<tr>
<td>13</td>
<td>4x PixelF(7:0)</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>4x TapG(15:8)</td>
<td>16 bit value representing the tap geometry (see section 9.4.5).</td>
</tr>
<tr>
<td>15</td>
<td>4x TapG(7:0)</td>
<td></td>
</tr>
</tbody>
</table>
| 16   | 4x Flags | Image flags: (8 bits, repeated 4 times)  
**Bits:** Usage:  
1:0 Interlacing:  
0x0 None (Progressive scan or line scan)  
0x1 Interlaced, first line after header from field 1 *  
0x2 Interlaced, first line after header from field 2  
0x3 Reserved for future use  
7:2 Reserved for future use, set to 0 |

* Field 1 is defined as the field that includes the top image line.
9.4.7.3  Arbitrary Line Marker

The line marker shall be as shown in the following table:

<table>
<thead>
<tr>
<th>Word</th>
<th>Content</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4x K28.3</td>
<td>Stream Marker.</td>
</tr>
<tr>
<td>2</td>
<td>4x 0x04</td>
<td>Arbitrary Line marker.</td>
</tr>
<tr>
<td>3</td>
<td>4x Xsize(23:16)</td>
<td>24 bit value representing the width in pixels of this line.</td>
</tr>
<tr>
<td>4</td>
<td>4x Xsize(15:8)</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4x Xsize(7:0)</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>4x Xoffs(23:16)</td>
<td>24 bit value representing the horizontal offset in pixels of this line</td>
</tr>
<tr>
<td>7</td>
<td>4x Xoffs(15:8)</td>
<td>with respect to the left hand pixel of the full Device image. Comment: Xoffs and Yoffs (in the header) tell the Host where this image is with respect to the full Device image size.</td>
</tr>
<tr>
<td>8</td>
<td>4x Xoffs(7:0)</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>4x DsizeL(23:16)</td>
<td>24 bit value representing the number of data words of this image line.</td>
</tr>
<tr>
<td>10</td>
<td>4x DsizeL(15:8)</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>4x DsizeL(7:0)</td>
<td></td>
</tr>
</tbody>
</table>

See Figure 34 for an example ROI.
10 Device Setup

This section covers Device discovery and bootstrap registers.

10.1 Device Discovery

Device discovery is initiated by the Host and consists of the following tasks:

- Match Device transmitter and Host receiver bit rates to obtain register access (section 10.1.2).
- Discover Devices and the connection topology (section 10.1.3).
- Detect the high speed upconnection (if supported) (section 10.1.4)
- Negotiate maximum packet sizes (section 10.1.5).
- Set the connection speed by reading ConnectionConfigDefault (section 10.1.6).

Discovered Devices need further configuration for operational use. Device configuration comprises bootstrap and Device register programming. Device configuration falls outside the scope of this specification.

Note: It is recommended that the implementation of the Device discovery process allows for users connecting and re-connecting cables while a system is powered up.

10.1.1 Connection State

Each connection has an associated connection state (Detected or Undetected) indicating the conditions defined in the following table:

<table>
<thead>
<tr>
<th>Connection state</th>
<th>Condition</th>
</tr>
</thead>
</table>
| Undetected       | Data reception not possible:  
|                  | • Host connection not physically connected to a Device.  
|                  | • Host connection physically connected to a Device but no low level connection lock. |
| Detected         | Data reception possible:  
|                  | • Host connection physically connected to a Device and low level connection lock achieved. |

Comment: Connection state detection is closely related to receiver bit clock lock, word alignment status, running disparity error flagging, etc. The implementation of connection state detection is implementation specific and outside the scope of this specification, however IDLE checking is recommended – see section 10.2.
10.1.2 Match Device and Host Bit Rates

Without prior knowledge the Device and Host connection scheme is unknown and the bit rates of the Device transmitters and Host receivers may not match. However the low speed Host to Device connection has a fixed standard bit rate allowing transfer of control packets from Host to Device. During Device discovery the Host shall use a maximum control packet size of 128 bytes (see section 10.1.5).

The process to match the downconnection bit rates is described below and shown in Figure 35.

- For each Host connection the Host shall write a connection reset command by writing the value 1 to the Device ConnectionReset bootstrap register.
- A Device receiving this connection reset command via its Master connection (connection 0) shall execute connection reset and activate its discovery connection configuration within 200ms (see section 10.3.28).
- The Host shall wait 200ms to allow for the Device to complete connection configuration.
- The Host shall initialize its receiver to the lowest discovery bit rate it supports (see section 4.3).
- The Host shall monitor the connection state.
- If the connection state is Detected within a timeout period then the Host and Device have established a connection.

Comment: A timeout period allows the Host receiver to attain lock and alignment. The required time is implementation specific and therefore not specified here.

- If the connection state is Undetected after the timeout period then the Host may repeat the process using another discovery bit rate.
Start

Reset connection by writing 1 to the ConnectionReset register (do not wait for ack)

Wait for 200 ms

Select one of the discovery bit rates and initialize Host receiver

Connection State

Detected

Done for this connection

Undetected

Timeout

Yes

No

Figure 35 — Matching Device and Host bit rates
10.1.3 Discover Devices and Connection Topology

The Host shall read the ConnectionConfigDefault register to find the number of expected connections. It shall then write to the ConnectionConfig register to enable the number of connections read from ConnectionConfigDefault. However it shall not change from the discovery rate at this stage.

Comment: The Host needs to know how many connections the Device requires. It is of little use if only three connections are connected on a camera that needs four connections – the Host needs to report that state as a discovery failure. Similarly the Host must not complete discovery after finding two connections, if the user is still connecting cables and four are needed.

If the Host does not support the number of connections specified by ConnectionConfigDefault, or the required number of connections cannot be discovered within a timeout period, then the Host shall maintain the master connection at the discovery speed.

Comment: This allows a user to reprogram ConnectionConfigDefault (if this is supported by the Device). The value of the timeout period is beyond the scope of this specification, and may be infinite.

Connection schemes may vary and can be as simple as a single link connection of one Device to one Host connection, or as complicated as multiple Devices with multiple connections connected to a many connection Host as shown in the following figure:

![Diagram of Device - Host connection schemes](image)

**Figure 36 — Examples of Device – Host connection schemes**

Note that a Host may not support all theoretically possible connection topologies.

Comment: If all connection topologies are supported, then the topology discovery process means that the user does not need to plug in cables in a specific order. However supporting a subset of topologies may allow a Host to be more cost-effective.

The first step in connection topology discovery is to find out for each Detected Host connection whether it is connected to a Device Master connection (connection 0) or an extension connection (connection > 0). This is accomplished by reading the DeviceConnectionID bootstrap registers for each Detected Host connection:

- Returned Device connection ID = 0: This Host connection goes to a Device Master connection.
- Returned Device connection ID > 0: This Host connection goes to a Device Extension connection.
An example is shown in the following figure:

![Connection diagram](image)

**Obtained information:**

<table>
<thead>
<tr>
<th>Host Connection</th>
<th>Connection State</th>
<th>Read DeviceConnectionId</th>
<th>Write MasterHostConnectionId</th>
<th>Read MasterHostConnectionId</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Detected</td>
<td>0 (Master)</td>
<td>HostID of A, (e.g. 1)</td>
<td>HostID of A (1)</td>
</tr>
<tr>
<td>B</td>
<td>Undetected</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>Detected</td>
<td>0 (Master)</td>
<td>HostID of C, (e.g. 2)</td>
<td>HostID of C (2)</td>
</tr>
<tr>
<td>D</td>
<td>Detected</td>
<td>1 (Extension)</td>
<td>HostID of A (1)</td>
<td>HostID of A (1)</td>
</tr>
<tr>
<td>E</td>
<td>Detected</td>
<td>2 (Extension)</td>
<td>HostID of A (1)</td>
<td>HostID of A (1)</td>
</tr>
<tr>
<td>F</td>
<td>Undetected</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

**Figure 37 — Connection topology example part 1**

The next step is to find out the connection topology: which extension connection belongs to which master connection.

For each Host connection connected to a Device Master connection the Host shall write a Host connection identification number to the `MasterHostConnectionID` bootstrap registers. The Host connection identification number uniquely identifies the connection within the Host context. Any 32 bit number is allowed except for 0 which is reserved to indicate an unknown Host ID.

Finally by reading the `MasterHostConnectionID` bootstrap registers via all Host Connections the connection scheme is known, as shown in the following figure:

```
Host

<table>
<thead>
<tr>
<th>Host Connection</th>
<th>Connection State</th>
<th>Read DeviceConnectionId</th>
<th>Write MasterHostConnectionId</th>
<th>Read MasterHostConnectionId</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Detected</td>
<td>0 (Master)</td>
<td>HostID of A, (e.g. 1)</td>
<td>HostID of A (1)</td>
</tr>
<tr>
<td>B</td>
<td>Undetected</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>Detected</td>
<td>0 (Master)</td>
<td>HostID of C, (e.g. 2)</td>
<td>HostID of C (2)</td>
</tr>
<tr>
<td>D</td>
<td>Detected</td>
<td>1 (Extension)</td>
<td>HostID of A (1)</td>
<td>HostID of A (1)</td>
</tr>
<tr>
<td>E</td>
<td>Detected</td>
<td>2 (Extension)</td>
<td>HostID of A (1)</td>
<td>HostID of A (1)</td>
</tr>
<tr>
<td>F</td>
<td>Undetected</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
```

**Figure 38 — Connection topology example part 2**
10.1.4 Detection of High Speed Upconnection

If the Host supports a high speed upconnection, it needs to discover if the Device also supports one, and if so, if there is a cable present to form the connection. A Host supporting a high speed upconnection shall therefore perform the following process.

- The Host shall read Discovery register \textit{HsUpconnection} via the low speed connection.
- If the value of bit 0 of the register is “1”, then the Host shall enable its high speed upconnection at the current downconnection speed and send IDLE words.
- The Host shall wait 200ms for the Device to achieve lock.
- When the Device achieves connection state Detected (i.e. bit and word lock) on its high speed upconnection, it shall switch all upconnection communications to the high speed connection from the low speed connection.
- The Host shall read Discovery register \textit{HsUpconnection} via the high speed connection. If the read succeeds, then the Host shall switch all upconnection communications to the high speed connection from the low speed connection. If the read fails then the Host shall disable its high speed upconnection, and continue to use the low speed upconnection.
- If the Device detects that the high speed connection gets disabled, it shall switch all upconnection communications back to the low speed connection.

\textbf{Comment:} Even though both Host and Device support a high speed upconnection, the cables in use may not. The Host can report this as an error, but this is beyond the scope of the specification.

\textbf{Comment:} To save power, the Device can detect the first read of \textit{HsUpconnection} and use that to enable its high speed upconnection receiver. It can also disable the receiver if it does not achieve lock.

\textbf{Comment:} Note that the Host is required to maintain IDLE words on the low speed upconnection even when it is using the high speed connection (see section 8.2.5.1). Therefore a switch back to the low speed connection is instantaneous.
10.1.5 Negotiate Maximum Packet Sizes

For each discovered Device the following packet sizes shall be negotiated:

- Control packet maximum size (see sections 8.6.4 and 10.3.31).
- Stream packet maximum size (see sections 8.5.2 and 10.3.32).

Control packet maximum size negotiation shall use the rules in the following table:

<table>
<thead>
<tr>
<th>Device</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>At all times:</strong></td>
<td><strong>Prior to Device discovery:</strong></td>
</tr>
<tr>
<td>• Shall support a control packet size of up to its maximum supported control packet size.</td>
<td>• Shall assume a maximum control packet size of 128 bytes.</td>
</tr>
<tr>
<td>• Shall report the maximum supported control packet size, which shall be at least 128 bytes, in the ControlPacketSizeMax bootstrap register.</td>
<td><strong>After discovering a Device:</strong></td>
</tr>
</tbody>
</table>

Stream packet maximum size negotiation shall use the rules in the following table:

<table>
<thead>
<tr>
<th>Device</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Connection reset:</strong></td>
<td><strong>After discovering a Device:</strong></td>
</tr>
<tr>
<td>• Shall initialize the bootstrap register StreamPacketSizeMax to 0 indicating &quot;not configured&quot;.</td>
<td>• Shall write the maximum stream packet size supported by the Host to the StreamPacketSizeMax bootstrap register.</td>
</tr>
<tr>
<td><strong>While StreamPacketSizeMax=0:</strong></td>
<td></td>
</tr>
<tr>
<td>• Shall not transmit stream packets.</td>
<td></td>
</tr>
<tr>
<td><strong>When StreamPacketSizeMax&gt;0 after a write from the Host:</strong></td>
<td></td>
</tr>
<tr>
<td>• Transmission of stream packets allowed with a packet size up to the maximum size set by the StreamPacketSizeMax bootstrap register.</td>
<td></td>
</tr>
</tbody>
</table>
10.1.6 Setting the Operating Bit Rate

The Host shall read the ConnectionConfigDefault register to find the required bit rate and number of connections operating at this bit rate, and shall then use the process in section 10.1.6.1 to set that rate and number of connections.

The bit rate and number of connections operating at this bit rate can also be changed at any point after Discovery that a new bit rate is needed on a link, in which case the process in section 10.1.6.1 shall be used. In that case the required new value of ConnectionConfig can be found from GenICam (see section 11) or system documentation.

All connections forming one link, including a high speed upconnection, shall operate at the same bit rate.

10.1.6.1 Setting the Bit Rate

- The Host shall write the required ConnectionConfig value to the Device and wait for the acknowledgement at the original connection speed.
- The Device shall acknowledge the ConnectionConfig access, then shall change all connections specified by ConnectionConfig to the specified bit rate. If a high speed upconnection is in use the Device shall change it to the specified bit rate.
- The Host shall write to its corresponding register(s) to set the corresponding connections to the same bit rate. If a high speed upconnection is in use the Host shall change it to the same bit rate.
- The Host shall wait 200ms to allow the Device to complete connection re-configuration.
- The Host shall monitor the connection state.
- If the connection state is Detected within a timeout period the Host and Device have re-established the connection.
- If the connection state is Undetected after a timeout period then the Host may repeat the process or return to the Discovery procedure.
- Similarly, if a high speed upconnection is in use:
  - The Device shall monitor the connection state.
  - If the connection state is Undetected after 200ms then it shall switch back to the low speed upconnection.
  - After 250ms the Host shall read Discovery register HsUpconnection via the high speed connection. If the read succeeds, the Host and Device have re-established the connection. If the read fails then the Host shall disable its high speed upconnection, and repeat the high speed upconnection discovery process (section 10.1.4).
See the following figure:

![Diagram showing the process of setting the bit rate](image)

**Figure 39 — Setting the bit rate**

### 10.2 Detection of Loss of Lock

It is recommended that the presence of IDLE is checked on each active connection at the rates specified in section 8.2.5.1. If IDLE is not seen for a consecutive number of occasions set by a threshold, it is recommended that first the receiver for that connection is reset (so e.g. character and word alignment re-established). If at the Host this does not recover lock, it is recommended that the Device is reset and the discovery process re-tried (i.e. on all connections).

**Comment:** The IDLE word both acts as a “heartbeat”, showing that the connection is functioning, and confirms that the receiver is correctly word aligned. The value for the threshold may be system-dependent, but should not be set so low that “burst” type errors, e.g. as a result of EMC/EMI type events, cause the connection to reset. There is some correlation between PoCXP events (see section 7) and loss-of-lock detection. E.g. if the Host switches off Device power then IDLEs will not be received and the connection will get reset, but note that if a Device is connected that does not use PoCXP, a connection will be established without any PoCXP event.

If the optional high speed upconnection loses lock, and the Device cannot recover lock, it shall switch back to the low speed upconnection. If the Host does not get acknowledges to high speed upconnection packets, it shall switch back to the low speed upconnection and repeat the high speed upconnection discovery process (section 10.1.4).


10.3 Device Memory Space and Bootstrap Registers

Device memory space allows both registers to control the Device, and larger blocks of memory such as for the XML file needed for GenICam. The start of Device memory space has defined bootstrap registers.

10.3.1 Strings

Strings stored in the bootstrap and manufacturer-specific registers space shall be NULL-terminated, encoded ASCII. The NULL terminator is included in the string length. However when a register stores a string which is the maximum length of that register, it does not have a NULL terminator.

Comment: This is standard GenICam behavior.

10.3.2 Feature Size

Reads and writes of bootstrap and manufacturer-specific registers with a size up to and including 104 bytes shall be implemented as one Control Command message.

Comment: This simplifies updating non-volatile memory in a Device, and ensures that most numerical values can be read or written in one access. 104 bytes is the payload size of the minimum value of ControlPacketSizeMax (128 bytes); therefore all CoaXPress products will support a read or write of this size.

Comment: In general it will be the role of the Host software to ensure that this requirement is met.

Device memory space shall be implemented so that reads and writes with a size over 104 bytes can be read or written either as one Control Command message, or as multiple word aligned messages of data size between 4 bytes and the defined register size, subject to the access not exceeding that allowed by ControlPacketSizeMax.

Comment: Large areas of Device memory such as XML files will in general exceed the maximum control packet size so reads and writes must be split into multiple accesses.

Note: Zipped XML files must be the correct size for the unzip process to work, and this size may not be a multiple of 4 bytes. Therefore Devices using zipped XML files, and all Hosts, shall support read accesses that are not a multiple of 4 bytes.

10.3.3 Miscellaneous Rules

Bootstrap registers shall be defined in the XML Device configuration file and respect the rules defined in SFNC (Standard Features Naming Convention) of the GenICam standard (see section 11).

All Device registers shall be 32-bit aligned, big-endian.

The Device shall not send a wait acknowledgement to an access to a bootstrap register. Unused bootstrap register bits shall be set to 0.

Manufacturer-specific registers can start after the bootstrap registers memory-space.

It is recommended that writable registers in the manufacturer-specific space are also readable.

Comment: This is better for high-reliability systems.
10.3.4 Bootstrap Registers

CoaXPress compliant Devices shall implement all mandatory bootstrap registers defined in this section.

The bootstrap registers are listed in Table 45, which uses the following indications:

**Group:**
- Support: This register is needed to support other standards, such as GenICam.
- GenICam: This register implements a GenICam feature of the same name.
- CXP: This is a register defined for the CoaXPress standard.

**Support:**
- M: Mandatory.
- O: Optional.
- X: Recommended to include in the Device’s XML file.

**Access:**
- R: Read and/or Write.
- W: Write.

Access modes given in parentheses are optional.

**Default value:**
- This is the value after a Connection Reset.

*Comment:* Note that a connection reset occurs at power up, so this is also the power-up value.

### Table 45 — Bootstrap registers

<table>
<thead>
<tr>
<th>Address</th>
<th>Name</th>
<th>Group</th>
<th>Support</th>
<th>Access</th>
<th>Length (bytes)</th>
<th>Register Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000000</td>
<td>Standard</td>
<td>Support</td>
<td>M</td>
<td>R</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>0x00000004</td>
<td>Revision</td>
<td>Support</td>
<td>M, X</td>
<td>R</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>0x00000008</td>
<td>XmlManifestSize</td>
<td>Support</td>
<td>M</td>
<td>R</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>0x0000000C</td>
<td>XmlManifestSelector</td>
<td>Support</td>
<td>M</td>
<td>R/W</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>0x00000010</td>
<td>XmlVersion</td>
<td>Support</td>
<td>M, X</td>
<td>R</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>0x00000014</td>
<td>XmlSchemaVersion</td>
<td>Support</td>
<td>M</td>
<td>R</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>0x00000018</td>
<td>XmlUrlAddress</td>
<td>Support</td>
<td>M</td>
<td>R</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>0x0000001C</td>
<td>iidc2Address</td>
<td>Support</td>
<td>M</td>
<td>R</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>0x00002000</td>
<td>DeviceVendorName</td>
<td>GenICam</td>
<td>M, X</td>
<td>R</td>
<td>32</td>
<td>String</td>
</tr>
<tr>
<td>0x00002020</td>
<td>DeviceModelName</td>
<td>GenICam</td>
<td>M, X</td>
<td>R</td>
<td>32</td>
<td>String</td>
</tr>
<tr>
<td>0x00002040</td>
<td>DeviceManufacturerInfo</td>
<td>GenICam</td>
<td>M, X</td>
<td>R</td>
<td>48</td>
<td>String</td>
</tr>
<tr>
<td>0x00002070</td>
<td>DeviceVersion</td>
<td>GenICam</td>
<td>M, X</td>
<td>R</td>
<td>32</td>
<td>String</td>
</tr>
<tr>
<td>0x00002B0</td>
<td>DeviceSerialNumber</td>
<td>GenICam</td>
<td>M, X</td>
<td>R</td>
<td>16</td>
<td>String</td>
</tr>
<tr>
<td>0x00002C0</td>
<td>DeviceUserID</td>
<td>GenICam</td>
<td>M, X</td>
<td>R/W</td>
<td>16</td>
<td>String</td>
</tr>
<tr>
<td>0x00003000</td>
<td>WidthAddress</td>
<td>CXP</td>
<td>M</td>
<td>R</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>0x00003004</td>
<td>HeightAddress</td>
<td>CXP</td>
<td>M</td>
<td>R</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>0x00003008</td>
<td>AcquisitionModeAddress</td>
<td>CXP</td>
<td>M</td>
<td>R</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>0x0000300C</td>
<td>AcquisitionStartAddress</td>
<td>CXP</td>
<td>M</td>
<td>R</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>0x00003010</td>
<td>AcquisitionStopAddress</td>
<td>CXP</td>
<td>M</td>
<td>R</td>
<td>4</td>
<td>Integer</td>
</tr>
</tbody>
</table>

*continued on next page …*
Bootstrap registers continued…

<table>
<thead>
<tr>
<th>Address</th>
<th>Name</th>
<th>Group</th>
<th>Support</th>
<th>Access</th>
<th>Length (bytes)</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00003014</td>
<td>PixelFormatAddress</td>
<td>CXP</td>
<td>M</td>
<td>R</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>0x00003018</td>
<td>DeviceTapGeometryAddress</td>
<td>CXP</td>
<td>M</td>
<td>R</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>0x0000301C</td>
<td>Image1StreamIDAddress</td>
<td>CXP</td>
<td>M</td>
<td>R</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>0x00003018 + &lt;n&gt; x 4</td>
<td>Image&lt;n&gt;-StreamIDAddress</td>
<td>CXP</td>
<td>M</td>
<td>R</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>0x00004000</td>
<td>ConnectionReset</td>
<td>CXP</td>
<td>M</td>
<td>W(R)</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>0x00004004</td>
<td>DeviceConnectionID</td>
<td>CXP</td>
<td>M</td>
<td>R</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>0x00004008</td>
<td>MasterHostConnectionID</td>
<td>CXP</td>
<td>M</td>
<td>R/W</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>0x0000400C</td>
<td>ControlPacketSizeMax</td>
<td>CXP</td>
<td>M</td>
<td>R</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>0x00004010</td>
<td>StreamPacketSizeMax</td>
<td>CXP</td>
<td>M</td>
<td>R/W</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>0x00004014</td>
<td>ConnectionConfig</td>
<td>CXP</td>
<td>M, X</td>
<td>R/W</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>0x00004018</td>
<td>ConnectionConfigDefault</td>
<td>CXP</td>
<td>M, X</td>
<td>R</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>0x0000401C</td>
<td>TestMode</td>
<td>CXP</td>
<td>M, X</td>
<td>R/W</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>0x00004020</td>
<td>TestErrorCountSelector</td>
<td>CXP</td>
<td>M, X</td>
<td>R/W</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>0x00004024</td>
<td>TestPacketCountTx</td>
<td>CXP</td>
<td>M, X</td>
<td>R/W</td>
<td>8</td>
<td>Integer</td>
</tr>
<tr>
<td>0x00004028</td>
<td>TestPacketCountRx</td>
<td>CXP</td>
<td>M, X</td>
<td>R/W</td>
<td>8</td>
<td>Integer</td>
</tr>
<tr>
<td>0x00004030</td>
<td>ElectricalComplianceTest</td>
<td>CXP</td>
<td>O</td>
<td>R/W</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>0x0000403C</td>
<td>HsUpConnection</td>
<td>CXP</td>
<td>M</td>
<td>R</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>0x00006000</td>
<td>Start of manufacturer-specific</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

These registers are described in the following sections.
10.3.5 Standard

This register shall provide a magic number indicating the Device implements the CoaXPress standard. The magic number shall be 0xC0A79AE5.

Comment: The magic number is an approximation of “CoaXPress”.

10.3.6 Revision

This register shall provide the revision of the CoaXPress specification implemented by this Device.

<table>
<thead>
<tr>
<th>Bit Offset (lsb &lt;&lt; x)</th>
<th>Width (bits)</th>
<th>Name</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>16</td>
<td>Major revision</td>
<td>v1.5 has major revision 1, therefore these bits hold the value 0x0001</td>
</tr>
<tr>
<td>0</td>
<td>16</td>
<td>Minor revision</td>
<td>v1.5 has minor revision 5, therefore these bits hold the value 0x0005</td>
</tr>
</tbody>
</table>

Therefore Devices compliant to this revision 1.1 of the specification shall return 0x00010001.

Comment: The sub-minor version is not coded in this register.

10.3.7 XmlManifestSize

This register shall provide the number of XML manifests available. At least one manifest shall be available.

10.3.8 XmlManifestSelector

This register shall select the required XML manifest registers. It shall hold a number between 0 and XmlManifestSize – 1.

Comment: A connection reset sets the value 0x00000000.

10.3.9 XmlVersion[XmlManifestSelector]

This register shall provide the version number for the XML file given in the manifest referenced by register XmlManifestSelector.

<table>
<thead>
<tr>
<th>Bit Offset (lsb &lt;&lt; x)</th>
<th>Width (bits)</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>8</td>
<td>Reserved</td>
<td>Shall be 0</td>
</tr>
<tr>
<td>16</td>
<td>8</td>
<td>XMLMajorVersion</td>
<td>The major version number of the XML file</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>XMLMinorVersion</td>
<td>The minor version number of the XML file</td>
</tr>
<tr>
<td>0</td>
<td>8</td>
<td>XMLSubMinorVersion</td>
<td>The sub-minor version number of the XML file</td>
</tr>
</tbody>
</table>
10.3.10  XmlSchemaVersion[XmlManifestSelector]

This register shall provide the GenICam schema version for the XML file given in the manifest referenced by register XmlManifestSelector.

<table>
<thead>
<tr>
<th>Bit Offset (lsb &lt;&lt; x)</th>
<th>Width (bits)</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>8</td>
<td>Reserved</td>
<td>Shall be 0</td>
</tr>
<tr>
<td>16</td>
<td>8</td>
<td>SchemaMajorVersion</td>
<td>The major version number of the schema used by the XML file</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>SchemaMinorVersion</td>
<td>The minor version number of the schema used by the XML file</td>
</tr>
<tr>
<td>0</td>
<td>8</td>
<td>SchemaSubMinorVersion</td>
<td>The sub-minor version number of the schema used by the XML file</td>
</tr>
</tbody>
</table>

10.3.11  XmlUrlAddress[XmlManifestSelector]

This register shall provide the address of the start of the URL string referenced by register XmlManifestSelector. This address shall be in manufacturer-specific register space (i.e above 0x6000). The string shall be in the format defined in by GenTL (see section 11).

10.3.12  Iidc2Address

For Devices that support the IIDC2 protocol (section 2.2 ref 6), this register shall provide the address of the start of the IIDC2 register space.

For Devices that do not support IIDC2, this register shall be set to the value 0x00000000.

**Comment:** IIDC2 can be supported to allow existing IIDC2 applications to be used.

**Comment:** Note that GenICam support is mandatory, therefore the Device needs an XML file that describes the IIDC2 registers. This XML file may be common to many IIDC2 products.

10.3.13  DeviceVendorName

This register shall provide the name of the manufacturer of the Device as a string.

10.3.14  DeviceModelName

This register shall provide the model name of the Device as a string.

10.3.15  DeviceManufacturerInfo

This register shall provide extended manufacturer-specific information about the Device as a string.
10.3.16 DeviceVersion
This register shall provide the version of the Device as a string.

10.3.17 DeviceSerialNumber
This register shall provide the serial number for the Device as a NULL-terminated string.
Comment: In v1.0 of the specification this was named DeviceID. It has been renamed because the GenICam feature it maps to has been renamed to DeviceSerialNumber.

10.3.18 DeviceUserID
This register shall provide a user-programmable identifier for the Device as a string. The Device shall provide persistent storage for this register so the value is maintained when power is switched off.

10.3.19 WidthAddress
10.3.20 HeightAddress
10.3.21 AcquisitionModeAddress
10.3.22 AcquisitionStartAddress
10.3.23 AcquisitionStopAddress
10.3.24 PixelFormatAddress
10.3.25 DeviceTapGeometryAddress
10.3.26 Image1StreamIDAddress
10.3.27 Image<n>StreamIDAddress
These registers shall provide the address in the manufacturer-specific register space of the use-case feature with the corresponding name (see section 11.2.1). Image<n>StreamIDAddress shall be set to 0 for stream numbers that are not supported by the Device.
Comment: This allows non-GenICam applications, or “black-box” type format converters, to support the standard use-case and allow continuous acquisition and display of images.

10.3.28 ConnectionReset
Writing the value 0x00000001 to this register shall provide Device connection reset. A Device receiving this connection reset command via its Master connection (connection 0) shall execute connection reset and activate its discovery connection configuration within 200ms. The Device shall clear the register back to 0x00000000 when it has activated its discovery connection configuration. The write by the Host should be regarded “fire and forget” without waiting for an acknowledgment.
Connection reset shall comprise:

- Activate the master connection. Extension connections shall not be activated.
- Initialize the bit rate for the master connection to the lowest discovery bit rate it supports (see section 4.3), with the corresponding value stored in ConnectionConfig.
- MasterHostConnectionID shall be set to 0x00000000, indicating unknown Host Connection ID.
- StreamPacketSizeMax shall be set to 0x00000000, indicating “not initialized”, and preventing any data packets being sent. The Device therefore sends just IDLE characters on the master connection.
- The stream control in the Device shall be reset so that the Packet Tag in the stream data packet shall start from 0, and for a multi-connection Device, the first stream packet sent shall be on connection 0.
- TestMode and TestErrorCountSelector shall be set to 0x00000000.
- All test mode counters shall be set to 0.
- ComplianceTest shall be set to 0x00000000.
- HsUpconnection shall be set to 0x00000000 if the Device does not support a high speed upconnection, or 0x00000001 if it does.
- Device trigger signal shall be set to 0.
- XmlManifestSelector shall be set to 0.

The Device shall also execute a connection reset after power-up.

Note: The Device shall ignore a connection reset received on a Device extension connection.

Comment: A Host does not initially know what it has been connected to, so it will send connection reset commands to all connections. The Device needs to ignore those on extension connections or discovery will fail.

Comment: In general it is not possible to read this register while it has the value 0x00000001.

10.3.29 DeviceConnectionID

This register shall provide the ID of the Device connection via which this register is read, as shown in Figure 37.

Comment: A Connection ID of 0 means that the connection is a Master connection. This is a static register, but with a different value depending on which connection it is read from.

10.3.30 MasterHostConnectionID

This register shall hold the Host Connection ID of the Host connection connected to the Device Master connection. The value 0x00000000 is reserved to indicate an unknown Host ID.

Note: The Device shall ignore a write to a Device extension connection.

Comment: This register is set by the Host to a system wide (Host) unique 32 bit ID number as part of the Device discovery process (see section 10.1.3). A connection reset sets the value 0x00000000.
10.3.31 ControlPacketSizeMax

This register shall provide the maximum control packet size the Host can read from the Device, or write to the Device. The size is defined in bytes, and shall be a multiple of 4 bytes. The defined size is that of the entire packet, not just the payload.

Comment: The control packet size is at least 128 bytes. See section 10.1.5.

10.3.32 StreamPacketSizeMax

This register shall hold the maximum stream packet size the Host can accept. The size is defined in bytes, and shall be a multiple of 4 bytes. The Device can use any packet size it wants to up to this size. The defined size is that of the entire packet, not just the payload.

Comment: This register is set by the Host as part of the Device discovery process. A connection reset sets the value 0x00000000. See section 10.1.5.

10.3.33 ConnectionConfig

This register shall hold a valid combination of the Device connection speed and number of active downconnections. Writing to this register shall set the connection speeds on the specified connections, and the high speed upconnection, if supported. If the new ConnectionConfig value results in a change of connection speed, the Device shall acknowledge the ConnectionConfig access at the original connection speed. Therefore it shall acknowledge the access before changing connection speed. See section 10.1.6.1.

Comment: Not all theoretical combinations of connection speed and number of connections may be usable. One register is used to ensure that the two variables are set simultaneously. The XML file and product documentation give valid combinations for the Device. A connection reset sets the value corresponding to the selected discovery rate and one connection.

<table>
<thead>
<tr>
<th>Bit Offset (lsb &lt;&lt; x)</th>
<th>Width (bits)</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>16</td>
<td>Number of connections</td>
<td>Number of activated Device downconnections (e.g. 1 for one connection, 2 for two connections, etc.)</td>
</tr>
<tr>
<td>0</td>
<td>16</td>
<td>Connection speed</td>
<td>Bit rate selection code, see Table 46.</td>
</tr>
</tbody>
</table>

Table 46 — Bit rate codes

<table>
<thead>
<tr>
<th>Bit Rate (Gbps)</th>
<th>Bit Rate Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.250</td>
<td>0x28</td>
</tr>
<tr>
<td>2.500</td>
<td>0x30</td>
</tr>
<tr>
<td>3.125</td>
<td>0x38</td>
</tr>
<tr>
<td>5.000</td>
<td>0x40</td>
</tr>
<tr>
<td>6.250</td>
<td>0x48</td>
</tr>
</tbody>
</table>
A write to ConnectionConfig shall reset the stream control in the Device so that the Packet Tag in the stream data packet shall start from 0, and for a multi-connection Device, the first stream packet sent shall be on connection 0. The stream control shall be reset even if the ConnectionConfig value is unchanged.

10.3.34 ConnectionConfigDefault

This register shall provide the value of the ConnectionConfig register that allows the Device to operate in its recommended mode.

Comment: The recommended mode would usually be the mode which gives the maximum performance.

If a Device can operate in more than one ConnectionConfig mode, it is recommended that a mechanism is provided via the manufacturer register space to allow a user to reprogram ConnectionConfigDefault.

Comment: This allows a user to e.g. set up a two connection camera to operate on one connection if that is appropriate for the user’s system. The expected mechanism would be “unlock / update / commit value” using registers in manufacturer-specific space to change ConnectionConfigDefault using a special utility program. See section 10.1.3.

10.3.35 TestMode

Writing the value 0x00000001 to this register shall enable test packets transmission from Device to Host. The value 0x00000000 shall allow normal operation. When the value is changed from 0x00000001 to 0x00000000 the Device shall complete the packet of 1024 test words currently being transmitted. See section 8.7.4.

Comment: A connection reset sets the value 0x00000000.

10.3.36 TestErrorCountSelector

This register shall select the required test count [TestErrorCountSelector] register. It shall hold a valid Device Connection ID 0 .. n-1, or n for the optional high speed upconnection.

Comment: A connection reset sets the value 0x00000000.

10.3.37 TestErrorCount[TestErrorCountSelector]

This register shall provide the current connection error count for the connection referred to by register TestErrorCountSelector.

Writing 0x00000000 to this register shall reset to zero the connection error count for the connection referred to by register TestErrorCountSelector.

Comment: A connection reset sets all connection test counters to zero. The error count is the number of incorrect words that have been received in test packets (see section 8.7.3).

10.3.38 TestPacketCountTx[TestErrorCountSelector]

This register shall provide the current transmitted connection test packet count for the connection referred to by register TestErrorCountSelector. See section 8.7.3.
Writing 0x00000000 to this register shall reset to zero the transmitted connection packet count for the connection referred to by register TestErrorCountSelector.

Comment: A connection reset sets all connection test counters to zero.

### 10.3.39 TestPacketCountRx[TestErrorCountSelector]

This register shall provide the current received connection test packet count for the connection referred to by register TestErrorCountSelector. See section 8.7.3.

Writing 0x00000000 to this register shall reset to zero the received connection packet count for the connection referred to by register TestErrorCountSelector.

Comment: A connection reset sets all connection test counters to zero.

### 10.3.40 ElectricalComplianceTest

If implemented, this shall be a non-volatile register to support formal compliance testing of the Device. It shall not be used at any other time. Writing the value 0x00000000 shall allow normal operation. Writing a valid ConnectionConfig value shall result in the following behavior when the Device is next powered up and no commands are sent on the upconnection.

- The connection speed and number of connections shall be set according to the value written to this register.
- Test packets as defined in section 8.7.2 shall be output on these connections.
- No other packets shall be sent on the connections other than Control Acknowledges.
- The Connector Indicator Lamps (section 5.4), if fitted, shall indicate compliance test mode.
- The Device shall not require any packets or IDLE words to be sent on the upconnection.

Comment: This allows the Device to be plugged into an analyzer for physical layer compliance testing. The Device can be setup for the bit rate to be tested, then when it is unplugged from the Host and plugged into the analyzer it will immediately output test data at the required bit rate. However when connected to a Host, the usual ConnectionReset access during discovery resets this register to 0x00000000, so any accidental access to this register (e.g. by a crashed Host PC) will automatically recover.

Comment: Alternatively, a manufacturer can supply a Device with dedicated firmware to allow physical layer compliance testing.

### 10.3.41 HsUpConnection

This register shall indicate Device support of the optional high speed upconnection.

<table>
<thead>
<tr>
<th>Bit Offset (lsb &lt;&lt; x)</th>
<th>Width (bits)</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>31</td>
<td>Reserved</td>
<td>Shall be 0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>HsUpConnectionSupport</td>
<td>Shall be 0 if a high speed upconnection is not supported. Shall be 1 if a high speed upconnection is supported.</td>
</tr>
</tbody>
</table>
11 GenICam Support

11.1 Introduction

CoaXPress products shall support GenICam. See section 2.1 for details of the GenICam standard.

On the Device side this means an XML file shall be provided with the Device description compatible with the GenApi module of GenICam. The bootstrap registers in the Device provide a means to access the XML file. The features of the Device shall follow GenICam’s SFNC if applicable, by using the name and type provided in the GenICam SFNC.

The Device vendor is free to use any suitable register layout in the Device, such as IIDC2 compliant or vendor specific, as long as an XML file is provided describing the layout and the features exposed, and providing the bootstrap registers defined in section 10.3.4 are supported.

To reduce the size of the XML file, a compressed version of the file can be stored in the Device. A compressed file shall use the DEFLATE and STORE methods of the ZIP format (section 2.2 ref 3).

Comment: This is mainly relevant to files stored in the Device, where non-volatile memory space may be limited. Note that the Device does not need to generate or understand the file, only store it.

On the Host side this means implementing a GenTL Producer to allow access to the Device’s XML file, provide a control interface and a streaming interface.

The following figure shows how GenICam (and in particular GenTL) terminology, shown in blue, maps to a CoaXPress Device and Host:

One of the goals of CoaXPress is to allow a user to buy a CoaXPress camera Device and easily interface it to any CoaXPress compliant Host (i.e. framegrabber and application software). In order to guaranty a
minimal level of interoperability, the CoaXPress specification addresses the use cases of the Device and Host to allow continuous acquisition and display of images.

Comment: The functionality defined in the “Use Case” is only a subset of what a CoaXPress Device can do.

11.2 Device Requirements for GenICam

CoaXPress compliant Devices shall implement the following GenICam versions:

- GenICam V2.3.1 or higher, plus
- SFNC V2.0 or higher

Comment: At the release date of this specification, there is not an overall GenICam release including SFNC v2.0. CoaXPress needs SFNC v2.0 to include the CoaXPress-specific features.

11.2.1 Use Case of Device

In addition to the bootstrap registers that are listed with “Support” = “X” in Table 45, all CoaXPress camera Devices shall support the features listed in Table 47 in their XML description files, with corresponding registers in the manufacturer-specific space. The table uses the following indications:

| Group: GenICam: | CXP: |
| This feature implements a GenICam feature of the same name. |
| This is a feature defined (or redefined) for the CoaXPress standard. |

Access: R: Read and/or Write. Access modes given in parentheses are optional.

<table>
<thead>
<tr>
<th>Name</th>
<th>Group</th>
<th>Access</th>
<th>Length (bytes)</th>
<th>Register Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Width</td>
<td>GenICam</td>
<td>R/(W)</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>Height</td>
<td>GenICam</td>
<td>R/(W)</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>AcquisitionMode</td>
<td>GenICam</td>
<td>R/(W)</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>AcquisitionStart</td>
<td>GenICam</td>
<td>W/(R)</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>AcquisitionStop</td>
<td>GenICam</td>
<td>W/(R)</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>PixelFormat</td>
<td>GenICam</td>
<td>R/(W)</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>DeviceTapGeometry</td>
<td>GenICam</td>
<td>R/(W)</td>
<td>4</td>
<td>Integer</td>
</tr>
<tr>
<td>Image&lt;n&gt;StreamID</td>
<td>CXP</td>
<td>R</td>
<td>4</td>
<td>Integer</td>
</tr>
</tbody>
</table>

Note: This list is only intended for camera Devices. Non-camera Devices do not have to support the list.

Comment: Bootstrap registers give the address of these registers to allow the Use Case to be supported by non-GenICam software. See section 10.3.19.

These features are described in the following sections.
11.2.1.1 Width
This feature provides the image width in pixels.

11.2.1.2 Height
This feature provides the image height in pixels.

11.2.1.3 AcquisitionMode
This feature controls the acquisition mode of the Device, as defined by the GenICam SFNC. For the mandatory use case, only “Continuous” needs to be supported, where images are captured continuously until stopped with an AcquisitionStop command.

11.2.1.4 AcquisitionStart
This feature starts acquisition from the Device, as defined by the GenICam SFNC. To allow the use of bootstrap register AcquisitionStartAddress, the register controlling this feature shall be single-shot, with the value 0x00000001 written to it to start acquisition.

*Comment:* If a Device has low level firmware that requires a value other than 0x00000001 to be written to its start register, then the Device will need to implement a virtual register that accepts the value 0x00000001 and then writes the required value to the start register.

11.2.1.5 AcquisitionStop
This feature stops acquisition from the Device, as defined by the GenICam SFNC. To allow the use of bootstrap register AcquisitionStopAddress, the register controlling this feature shall be single-shot, with the value 0x00000001 written to it to stop acquisition.

*Comment:* If a Device has low level firmware that requires a value other than 0x00000001 to be written to its stop register, then the Device will need to implement a virtual register that accepts the value 0x00000001 and then writes the required value to the stop register.

*Comment:* Note that the SFNC defines that AcquisitionStop should complete transmission of the current image.

11.2.1.6PixelFormat
This feature provides the pixel format. The value is defined by the GenICam PFNC. It is the responsibility of the Device to map this to a valid PixelF code to send over CoaXPress. See section 9.4.1.

11.2.1.7 DeviceTapGeometry
This feature provides the tap geometry. The value is defined by the GenICam SFNC. It is the responsibility of the Device to map this to a valid TapGeometry code to send over CoaXPress.

11.2.1.8 Image<n>StreamID
This gives the Stream ID of the nth image stream from the Device. e.g. Image2StreamID gives the ID for the second stream. It is recommended that the primary image stream from the Device (Image1StreamID) has a stream ID of 0 (see section 9.3).
Comment: Multiple stream ID features are part of the use case to allow for display of images from any Device with more than one stream, e.g. a two-tap camera Device.

11.2.2 XML File Location

The XML file can be stored in two types of location, and shall be provided in at least one of them:

- Non-volatile memory in the Device (recommended).
- The Device vendor’s website.

The `XmlManifestTable` provides a method to access multiple options of the XML file, both to allow for different locations, and for different versions of the XML file and/or GenApi.

The URL format shall be as specified in the GenTL specification. Example URLs are shown below.

Example: “Local: MyFilename.zip;B8000;33A?SchemaVersion=1.0.0” is a ZIP file starting at address 0xB8000 in the Device with a length of 0x33A bytes. The filename is “MyFilename.zip”, and the XML file uses GenICam schema v1.0.0.

Comment: The filename is not used here, but may be useful when comparing the file with other versions on the Host system.


Note: The Host does not know the length of the string, so it must read it 4 bytes at a time until it sees the NULL terminator.

11.3 Host Requirements for GenICam

CoaXPress compliant Hosts shall implement the following GenICam versions:

- GenICam V2.3.1 or higher, plus
- SFNC V2.0 or higher
- GenTL SFNC V1.0 or higher

Comment: At the release date of this specification, there is not an overall GenICam release including SFNC v2.0 or the GenTL SFNC. CoaXPress needs these to include the CoaXPress-specific features.

CoaXPress compliant Hosts shall provide a GenICam GenTL Producer API module.

This GenTL Producer shall implement the acquisition engine to enable image streaming from camera to host, as well as the Port functionality. The Host shall therefore provide a set of XML files to describe its features (system, interface, device, datastream).

Comment: The XML files would typically be embedded into the GenTL Producer.
11.4 Use Case of Host

The features required for a GenICam GenTL Producer as described in the previous section provide the Use Case for the Host.
Annex A: Detailed Cable Specification

A.1 Introduction

This annex defines detailed cable requirements needed by cable manufacturers to design and verify a cable. It should be read in conjunction with section 5. Whether a given cable will work reliably depends on its DC and AC frequency behavior.

This annex is also needed by a CoaXPress user who chooses to make their own cables. They need to choose a 75 Ω coaxial cable type, cut it to the desired length and terminate it with CXP-connectors. In that way, a large set of coaxial cables and coaxial cable types can be used, including legacy ones. These cables are field-installed and are called custom-cables. It shall then be the responsibility of the installer to check whether the system has enough operational margin for the given application.

A.2 Impedance

The cable used in CXP-cables and in the separate lanes of a CXP-multi-cable shall be coaxial with a characteristic impedance of 75 Ω ± 4 Ω. The connectors used in CXP-cables and CXP-multi-cables shall be of the 75 Ω type.

Comment: Impedance errors in connectors are allowed for in the return loss requirements.

When a series coupling of CXP-cables is considered, all of the connectors used have to be of the 75 Ω type, including any inline couplers.

A.3 Return Loss of Cables

A CXP-cable and the separate lanes of a CXP-multi-cable shall have a return loss better than or equal to the one given in the following table:

<table>
<thead>
<tr>
<th>Frequency range</th>
<th>Return Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 - 500 MHz</td>
<td>-20 dB</td>
</tr>
<tr>
<td>500 MHz – 3.2 GHz</td>
<td>-15 dB</td>
</tr>
</tbody>
</table>

A.4 Cable Length Limitations

The maximum length of a CoaXPress cable is the lowest figure from three different requirements: power supply voltage drop, high speed connection requirements and low speed connection requirements.

A.4.1 Power Supply Voltage Drop

A CXP-cable and the separate lanes of a CXP-multi-cable shall each have a total DC roundtrip resistance of less than 4.98 Ω for each of the coax cables.
Comment: As explained in section 7.3.2.1, there is for the CXP-cable a total voltage drop budget of 3.5V. In the worst case current condition of 703mA (see section 7.3.2.2), this leaves the CXP-cable with an allowed roundtrip resistance of 4.98Ω. This roundtrip resistance includes the connector resistances, the resistances of the core of the coax and of the shield of the coax. In practice, most of this resistance is determined by the core of the cable, however, the shield resistance shall not be neglected.

A.4.2 High Speed Connection Requirements

A CXP-cable and the separate lanes of a CXP-multi-cable that are specified for a given bit rate shall have an attenuation that is less or equal to the attenuation in the following table at its corresponding frequency.

Annex A: Table 2 — Specification of the maximum attenuation for a cable

<table>
<thead>
<tr>
<th>Bit Rate (Gbps)</th>
<th>Maximum Attenuation (dB)</th>
<th>@ Frequency (GHz)</th>
<th>Belden 1694A (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.250</td>
<td>-21.2</td>
<td>0.6250</td>
<td>130</td>
</tr>
<tr>
<td>2.500</td>
<td>-26.0</td>
<td>1.2500</td>
<td>110</td>
</tr>
<tr>
<td>3.125</td>
<td>-26.8</td>
<td>1.5625</td>
<td>100</td>
</tr>
<tr>
<td>5.000</td>
<td>-20.9</td>
<td>2.5000</td>
<td>60</td>
</tr>
<tr>
<td>6.250</td>
<td>-15.8</td>
<td>3.1250</td>
<td>40</td>
</tr>
</tbody>
</table>

Comment: As an example the length of Belden 1694A cable that can be associated with that attenuation level at the given frequency is also shown in this table.

A series coupling of cables, possibly using inline couplers, shall have an attenuation in total of less than, or equal to, the attenuation in this table.

A.4.3 Low Speed Connection Requirements

A CXP-cable and the separate lanes of a CXP-multi-cable shall have a signal attenuation at 30 MHz of less than, or equal to, -4.74dB.

Comment: The performance of the low speed connection at 20.83 Mbps is determined on how well the edges are preserved during transmission. Edge preservation is mainly based on preserving the third harmonic component during transmission. In an 8B/10B coded system at 20 Mbps, the third harmonic is located at 30 MHz. The attenuation of -4.74dB @ 30 MHz equals to the attenuation of a 130m Belden 1694A cable.

A.5 Cable Current Capacity

A CXP-cable and the separate lanes of a CXP-multi-cable shall each be designed to carry 1A in normal operation.
A.6 Cable Attenuation

Comment: The receiver circuit in the Host (HT) includes an equalizer expecting to receive a signal that has suffered from frequency dependent losses that are characteristic for a coaxial cable, with skin-effect losses and possibly also dielectric losses. Therefore the attenuation characteristic of a CXP-cable has to closely resemble that of an ideal coaxial cable having only skin-effect and dielectric losses. This is referred to as "cable-like behavior". Using curve fitting it can be determined whether the cable attenuation characteristic of a possible CXP-cable is close enough to such an ideal cable.

A CXP-cable and the separate lanes of a CXP-multi-cable shall have an attenuation versus frequency characteristic exhibiting cable-like behavior over the frequency ranges indicated in the following table. A series coupling of cables shall also fulfill this requirement as if it is one cable including all of its connectors and inline couplers.

Annex A: Table 3 — Frequency range in which cable-like behavior is to be guaranteed

<table>
<thead>
<tr>
<th>Cable Rating (Gbps)</th>
<th>Frequency Range from (MHz) to (GHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.25</td>
<td>1 to 0.6250</td>
</tr>
<tr>
<td>2.5</td>
<td>1 to 1.2500</td>
</tr>
<tr>
<td>3.125</td>
<td>1 to 1.5625</td>
</tr>
<tr>
<td>5</td>
<td>1 to 2.5000</td>
</tr>
<tr>
<td>6.25</td>
<td>1 to 3.1250</td>
</tr>
</tbody>
</table>

The following formula shall be used for fitting, where \( f \) is the frequency in Hz:

\[
\text{Attenuation (dB)} = -A_0 - \text{SQRT}(f / A_1) - f / A_2 \quad (\text{formula 1}^3)
\]

This formula has three fitting parameters:

- \( A_0 \) is linked with DC-attenuation.
- \( A_1 \) is linked with skin-effect losses.
- \( A_2 \) is linked with possible dielectric losses in the cable.

The fitting is based on maximum likelihood estimation, with equal weights for all measurements. During the fitting process, \( A_0, A_1, \) and \( A_2 \) are free to change.

For fitting, between 40 and 50 measurement points for each decade shall be used, nominally logarithmically spaced over the entire frequency range.

The maximum difference between the measurement and the fitted attenuation curve shall never be more than 1 dB.

---

3 As an example, for 100 meter Belden 1694A, the coefficients are \( A_0=0.379; A_1=2.92 \times 10^6; A_2=4.811 \times 10^8 \) in agreement (+-0.15 dB) with the manufacturer’s table in section 2.2 ref 2.
Example: Two 10m cables, coupled together with a 50 Ω (instead of the correct 75 Ω) inline coupler, are measured in the lab from 1 MHz to 3.125GHz:

Annex A: Figure 1 — Deviation from normal attenuation behavior versus frequency

This cable combination has a residue of more than the 1 dB between 3 GHz and 3.5 GHz (as a result of the incorrect 50 Ω inline coupler!). Therefore it does not pass at a bit rate of 6.25 Gbps, however despite the 50 Ω connector it does exhibit cable-like behavior at 5 Gbps and lower bit rates.

A.7 Multi-Cable Requirements

A CXP-multi-cable shall have a 1:1 mapping between the connectors at each end of the cable, i.e. the connection order shall not be crossed within the cable.
The following figure shows the rules for the connector on a CXP-multi-cable (dimensions in mm):

Annex A: Figure 2 — CXP-multi-cable rules

Note that this figure shows an example connector design. Cable vendors can use any design of housing that accepts DIN 1.0/2.3 connectors and complies with the dimensions shown.

A CXP-multi-cable shall be marked to indicate the master connection at each end of the cable.

Comment: The arrow shown in the figure above is an example of suitable marking.

The 1mm (min) slot in the above figure can be used to provide an additional mechanical locking function by preventing the release tab from being moved. This can be a tab supplied with the cable, or for critical applications it allows the Device or Host vendor to supply a bracket that fixes to the Device or Host and engages in the slot. See section 5.2.2.
Annex B: Chipset Requirements

B.1 Introduction

This annex defines electrical requirements needed by chipset manufacturers to design and verify new transceivers. It should be read in conjunction with section 6.

Some of this annex may also be useful to Device and Host manufacturers as part of product compliance testing.

The high speed connection is usually the downconnection from the Device to the Host, and the low speed connection is in the opposite direction, so usually the upconnection from the Host to the Device. In systems using the optional high speed upconnection these are reversed, but for clarity this annex describes the usual case. A high speed upconnection shall implement a Host Transceiver in the Device, and a Device Transceiver in the Host according to this annex.

B.2 Top Level Overview

Annex B: Figure 1 (duplicated from section 6.2 for ease of reference) shows the electrical block diagram of one connection in a CoaXPress system, with the Device on the left connected to the Host on the right by a coax cable with a characteristic impedance of 75 \( \Omega \). The figure also defines the test points Tp1 to Tp4 used in this section.

The high speed connection is usually the downconnection from the Device to the Host, and the low speed connection is in the opposite direction, so usually the upconnection from the Host to the Device.

![Electrical block diagram](image)

**Comment:** Voltages shown at the output of the PRU and the input to the PTU are typical examples.

Connected to the Phy \( \Phi_D \) in the Device there shall be a Device Transceiver (DT) circuit that transmits and receives the bidirectional full duplex data over the coax cable. The Device Transceiver circuit shall split the incoming low-speed data signal and the outbound high speed data signal. The low speed data
will have lost some of its lower frequency components, resulting in baseline wandering, so the Device Transceiver circuit shall compensate for this, restoring the signal to the original digital bit stream at Tp1.

The I/O pin of the Device Transceiver circuit shall be coupled to the coax cable through a capacitor \( C_d \) that AC couples the data between this I/O pin and the center contact of the coax connector.

Connected to the Phy \( \Phi_H \) in the Host there shall be a Host Transceiver (HT) circuit that receives and transmits the bidirectional full duplex data over the coax cable at the Host side. The Host Transceiver circuit shall split the incoming high-speed data signal and the outbound low-speed data signal.

The incoming high-speed data signal may have suffered from frequency dependent attenuation in the coax cable, therefore the Host Transceiver circuit shall also equalize the incoming signal and restore the analog signal to the original digital bit stream, presented at Tp4 to the Phy \( \Phi_H \).

The I/O pin of the Host Transceiver circuit shall be coupled to the coax cable through a capacitor \( C_d \) that AC couples the data between this I/O pin and the center contact of the coax connector.

Systems using Power over CoaXPress (PoCXP) shall also implement a Power Transmit Unit (PTU) or Power Receive Unit (PRU), as described in section 7 on PoCXP.

B.3 High Speed Connection

B.3.1 Cable Driver

The high speed connection Device Transceiver requirements are specified by the following eye diagram at Tp2:

![Eye Diagram](image)

Annex B: Figure 2 — Definition of the parameters in the transmit EYE at Tp2
This eye defines the normative parameters measured at Tp2, at the Device connector when in the test state (see section 8.7) outputting 8B/10B test data. The transmit amplitude, the maximum rise and fall times, and the maximum amount of jitter shall be as indicated in the following table:

### Annex B: Table 1 — High speed connection parameters at the transmit (Device) side

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Units</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Delta V_{TX} )</td>
<td>Transmit amplitude</td>
<td>450</td>
<td>600</td>
<td>700</td>
<td>mV</td>
<td>At Tp2 into 75Ω.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>At Tp2 into 50Ω. See note 1 below.</td>
</tr>
<tr>
<td>( \Delta V_{EYE}/\Delta V_{TX} )</td>
<td>Relative EYE opening</td>
<td>0.7</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>At Tp2 into 75Ω or 50Ω. See note 3 below.</td>
</tr>
<tr>
<td>( t_{rise}, t_{fall} )</td>
<td>Rise, fall time</td>
<td>as short as possible</td>
<td>90</td>
<td>ps</td>
<td>At Tp2 into 75Ω or 50Ω, between 20% and 80% of amplitude. See note 3 below.</td>
<td></td>
</tr>
<tr>
<td>( T_j )</td>
<td>Transmit jitter</td>
<td>as low as possible</td>
<td>20</td>
<td>%UI</td>
<td>At Tp2 into 75Ω or 50Ω. See notes 2 and 3 below.</td>
<td></td>
</tr>
</tbody>
</table>

**Comment:** Jitter is defined with respect to the Unit interval, resulting in relaxed normative absolute parameters for lower speed Devices.

**Note 1:** This parameter is quoted at 50 Ω to allow practical measurement with high speed oscilloscopes which will have a 50 Ω input. The remaining parameters are quoted at 75 Ω because these can be measured using a high-speed converter. The 50 Ω figure assumes that the driver has an internal 75 Ω series terminator.

**Note 2:** The jitter figure \( T_j \) is a system figure comprising the combination of jitter from both the Phy \( \Phi_D \) and the Device Transceiver (DT). To allow practical implementation of the Phy \( \Phi_D \), the jitter contribution from the Device Transceiver should be minimized.

**Note 3:** The best method of testing these parameters is specified in the Compliance Test Specification document (section 2.2 ref 8). That document will determine whether to test into 50 Ω or into 75 Ω, and decide on all the specific elements in the test set-up. This may even be dependent on the connector type.

The high speed connection Device Transceiver shall not apply pre-emphasis or de-emphasis.

**Comment:** Pre-emphasis would make it more difficult for the cable receiver to do its job, because the combination of pre-emphasis and cable attenuation may result in an attenuation / frequency plot that does not have cable-like behavior. See section A.6.
B.3.2 Cable Receiver

At the receive side, the Host shall be able to receive the 8B/10B signals from a compliant Device over a length of the Belden 1694A (section 2.2 ref 2) coaxial cable having the attenuation given in the following table at the indicated frequency for the bit rates supported by the Host:

Annex B: Table 2 — High speed connection parameters at the receiver (Host) side

<table>
<thead>
<tr>
<th>Bit Rate (Gbps)</th>
<th>Attenuation (dB)</th>
<th>@ Frequency (GHz)</th>
<th>Equivalent Belden 1694A (m)</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.250</td>
<td>-22.0</td>
<td>0.625</td>
<td>135 m</td>
<td>8B/10B coding, full ΔV_{TX} range, power transmission &amp; low speed connection active, BER &lt;10^{-12}</td>
</tr>
<tr>
<td>2.500</td>
<td>-27.2</td>
<td>1.250</td>
<td>115 m</td>
<td>&quot;</td>
</tr>
<tr>
<td>3.125</td>
<td>-28.1</td>
<td>1.5625</td>
<td>105 m</td>
<td>&quot;</td>
</tr>
<tr>
<td>5.000</td>
<td>-22.6</td>
<td>2.500</td>
<td>65 m</td>
<td>&quot;</td>
</tr>
<tr>
<td>6.250</td>
<td>-17.8</td>
<td>3.125</td>
<td>45 m</td>
<td>&quot;</td>
</tr>
</tbody>
</table>

Adaptive equalization shall restore the original bit-stream, followed by the lock of the clock and data recovery in the Phy Φ_H. When a Host is able to receive at different bit rates, for each of these bit rates, it shall be able to receive through a length of Belden 1694A cable having the attenuation given this table at the indicated frequency.
B.4 Low Speed Connection

B.4.1 Baseline Wandering

Due to the high-pass filtering induced by \( Z_p \) at the Host, the signal measurable at Tp3 will show baseline wandering as shown in the following figure:

Annex B: Figure 3 — Signal transmitted at Tp3 by the Host showing baseline wander

This baseline wandering shall be taken care of in the Device Transceiver circuit.

Comment: By fixing the inductive part of \( Z_p \) to 11.5 \( \mu \)H (see section 6.5), base wander shall be present in a predictable way. In principle this base wander can be diminished by using an inductor of 40 \( \mu \)H or more. However, inductors of that value (and that also resist higher current levels), are physically large components! The 11.5 \( \mu \)H value is a nice compromise, leading to a physically small inductor, with a baseline wander that can still be restored by the receiver in the Device Transceiver circuit. Using a fixed value also makes interoperability more easily achievable since a known baseline wander can easily be restored.
B.4.3 Low Speed Cable Driver

The low speed connection cable driver requirements (part of the Host Transceiver) are specified by the following eye diagram at Tp3:

![Eye diagram of low speed transmit EYE at Tp3](image)

Annex B: Figure 4 — Definition of the parameters in the low speed transmit EYE at Tp3

This eye defines the normative parameters measured at Tp3, at the Host connector when outputting 8B/10B test data. The transmit amplitude, the minimum and maximum rise and fall times, and the maximum amount of jitter are indicated in the following table:

Annex B: Table 3 — Low speed connection parameters at the transmit (Host) side

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Parameter Description</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Units</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ΔV_{TXLF}</td>
<td>Transmit amplitude</td>
<td>90</td>
<td>130</td>
<td>180</td>
<td>mV</td>
<td>At Tp3 into 75Ω.</td>
</tr>
<tr>
<td>ΔV_{EYELF}/ΔV_{TXLF}</td>
<td>Relative EYE opening</td>
<td>0.75</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>At Tp3 into 75Ω.</td>
</tr>
<tr>
<td>t_{riseLF}, t_{fallLF}</td>
<td>Rise, fall time</td>
<td>5</td>
<td>11</td>
<td>20</td>
<td>ns</td>
<td>At Tp3 into 75Ω, between 20% and 80% of amplitude.</td>
</tr>
<tr>
<td>T_{JLF}</td>
<td>Transmit jitter</td>
<td>-</td>
<td>as low as possible</td>
<td>5</td>
<td>ns</td>
<td>At Tp3 into 75Ω. See note below.</td>
</tr>
</tbody>
</table>

*Comment:* These parameters are quoted at 75 Ω because at this speed it is easy to make measurements with a 75 Ω terminator and a high impedance oscilloscope input.

*Comment:* The definition of the minimum rise and fall times are important, since unneeded steepness must be avoided to achieve maximum crosstalk tolerance with the high speed connection data.

*Note 1:* The Power Transmitting Unit (PTU) may be present in the Host, and connected through Z_p. During this measurement it is however not required to use the Power Transmitting Unit for providing power to the Device.
Note 2: To verify the correct eye pattern, signal processing is needed, either in the oscilloscope or off-line, to equalize the low-frequency loss due to the impedance $L_p$ or $Z_p$ must be disconnected.

Note 3: For clarity the eye diagram is shown with no simultaneous flow of high speed connection data since that would corrupt the EYE parameters severely.

Note 4: The jitter figure $T_{jLF}$ is a system figure comprising the combination of jitter from both the Phy $Φ_H$ and the Host Transceiver (HT). To allow practical implementation of the Phy $Φ_H$, the jitter contribution from the Host Transceiver should be minimized.

B.4.4 Low Speed Cable Receiver

At the receive side, the Device shall be able to receive the 8B/10B signals from a compliant Host over a minimum length of the well-specified Belden 1694A coaxial cable.

Annex B: Table 4 — Low speed connection parameters at the receive (Device) side

<table>
<thead>
<tr>
<th>Bit Rate (Mbps)</th>
<th>Attenuation (dB)</th>
<th>@ Frequency (MHz)</th>
<th>Equivalent Belden 1694A (m)</th>
<th>Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>20.83 J</td>
<td>-4.9</td>
<td>30</td>
<td>135 m</td>
<td>8B/10B coding, full $ΔV_{TXLF}$ range power transmission &amp; high speed connection active, BER &lt; $10^{-12}$</td>
</tr>
</tbody>
</table>

B.5 Guidance Notes

Given that good PCB layout and component choice is critical to achieve CoaXPress performance (see section 6.8), and for many firms CoaXPress will be the highest speed design they have implemented, it is recommended that chipset manufacturers provide:

- Sample PCB artwork.
- Detailed layout guidance.
- Component choice recommendations.
Annex C: Descriptive Clause

C.1 Background of Standardization

C.1.1 Background of the Original CoaXPress Technology

The main technologies behind CoaXPress were developed as a part of MEDEA+ projects ASIC-CCD (2A206) and TritonZ (2A207). The MEDEA+ program is a EUREKA cluster focusing on micro and nanoelectronics, which are key enabling technologies for electronics and Information and Communications Technology (ICT).

The MEDEA+ label and the funding of the R&D by the local authorities in Belgium and the Netherlands were key for making what today is called CoaXPress.

C.1.2 Handover from the CoaXPress Consortium to JIIA

JIIA launched a next-generation interface working group at its inception in 2006, and the group had been studying two methods (a Pixel Clock method and a Packet method) for next-generation high-speed interface system.

In 2007, AIA (formerly Automated Imaging Association) based in the North America, EMVA (European Machine Vision Association) based in Europe and JIIA created the “Cooperation Agreement on Global Coordination of Machine Vision Standardization” in order to avoid the duplication of similar standards development.

In June 2009, the CoaXPress Consortium proposed that JIIA became the organization for the Global Standardization of a high-speed interface using a coaxial 75 Ω cable named “CoaXPress” and developed by the Consortium, and then in September 2009 JIIA accepted this proposal. In December 2009 the CoaXPress Consortium edition of the specification was handed over to JIIA, then the CoaXPress Standard was created by the JIIA CoaXPress Working Group.
C.2 Version Changes

C.2.1 v1.0 to v1.1

v1.1 adds support for CXP-multi-connectors and CXP-multi-cables based on the DIN 1.0/2.3 connector (see sections 5.2 and A.7). It also allows for a high speed upconnection to allow very fast data transfer and triggers to a Device (see section 10.1, in particular section 10.1.4). These additions are part of the roadmap to allow CoaXPress to remain at the leading edge of performance. v1.1 is English-only; a Japanese translation may be released in future.

Note that v1.1 Hosts additionally need to support some v1.0 features to allow operation with v1.0 Devices, and v1.1 Devices additionally need to support some v1.0 features to allow operation with v1.0 Hosts.

v1.1 also includes the following key changes:

- One coax cable – a “Link” – renamed to “Connection”. A master connection and extension connections now form a “Link”. This is for consistency with the GenICam SFNC, and results in the renaming of some bootstrap registers. See section 4.1.
- Some changes to the connector indicator lamp indications to improve clarity. See section 5.4.
- Slightly relaxed physical layer parameters (identical to the Specification Change Notice for v1.0 published by JIIA in document JIIA CXPR-004-2012). See section 6 and Annex B.
- Additional requirements for PoCXP in Devices: A startup/brownout current limit is defined; a discharge time for $C_s$ is defined; clarification of minimum load power requirements for multi-connection Devices. See sections 7.3.2.2 and 7.3.3.2.
- A simplified packet priority scheme. See section 8.2.4.
- GPIO has been removed to allow a future version of CoaXPress to implement an event packet using K28.0. Note that this event packet could include GPIO as one event type. See section 8.3.1.
- Tentative acknowledgements have been replaced by a wait acknowledgement, which allows the Host to stay in control of the command. See section 8.6.1.
- Improvements to the link test system: New counters are required to show the number of packets sent and received, including new bootstrap registers in the Device. See section 8.7.
- The mapping of the GenICam PixelFormat feature to the CoaXPress PixelF value is now fully defined. See sections 9.4.1.1 and 11.2.1.6.
- Rewording of the discovery process, with additional requirements defined for bootstrap register $LinkConfigDefault$ (now called $ConnectionConfigDefault$) for both the Host and Device, and a changed value of $LinkConf$ (now called $ConnectionConf$) following $LinkReset$ (now called $ConnectionReset$). See sections 10.1, 10.3.28 and 10.3.34.
- Additional Device memory space access requirements defined. See section 10.3.
• Additional bootstrap registers defined to allow support of the basic use case without the need for GenICam. See sections 10.3.19 to 10.3.27. This also results in new requirements for the features AcquisitionStart and AcquisitionStop. See sections 11.2.1.4 and 11.2.1.5.

• Additional optional bootstrap register to more easily support electrical compliance test for a Device. See section 10.3.40.

• Much of the GenICam sections have been deleted, as these the requirements are now fully defined in the GenICam (in particular GenTL) specification.

• GenICam requirements merged into one section, section 11.

v1.1 also includes detail wording changes to clarify the meaning of the specification, in particular:

• Improved wording of CRC requirements. See section 8.2.2.2.

• StreamPacketSizeMax (formerly StreamPacketDataSize) and ControlPacketSizeMax (formerly ControlPacketDataSize) now clearly stated to be in units of bytes, and to apply to the entire packet size. See sections 10.1.5, 10.3.31 and 10.3.32.

• Bootstrap registers specified using the “lsb << x” notation, for consistency with GenCP.

• Improved wording of the impedance requirements for CXP-cables. See section A.2.

C.2.2 v1.1 to v1.1.1

v1.1.1 has a limited number of changes to clarify some points in the v1.1 specification. The main clarifications are:

• Clarification that the Packet Tag and connection order should only be reset as part of a ConnectionReset or a write to ConnectionConfig. See sections 8.5.3, 8.5.5 and 10.3.33.

• In the Control Transactions section, the rules cross-reference the existing requirement in section 10.3.3 not to use Wait Acknowledgements for bootstrap registers, plus the wording has been improved to make it clearer that only one final acknowledgement should be sent. See section 8.6.1.1.

• The Pixel Types section has been reworded following changes in the PFNC. Devices are no longer required to use the “pmsb” suffix for packed pixel formats, which many vendors had ignored anyway. See section 9.4.1.

• The Discovery section clarifies the point at which a Host is expected to enable extension connections. See sections 10.1.3 and 10.1.6.

• The Discovery section also shows revised versions of Figure 37 and Figure 38 which show a Device with more than two connections, to clarify that DeviceConnectionID can be greater than 1 (as already defined by section 10.3.29).

• The bootstrap registers that should be listed in the XML file are now indicated. See section 10.3.4.
C.3  Members Involved

C.3.1  The Main Authors of the Original CoaXPress Specification

The main authors of the original CoaXPress standard were:

- Chris Beynon  
  Active Silicon Ltd.
- René Weltje  
  Adimec Advanced Image Systems B.V.
- Maarten Kuijk  
  EqcoLogic N.V.

In addition, the following people contributed to original CoaXPress specification:

- Martin Bone  
  Active Silicon Ltd.
- Colin Pearce  
  Active Silicon Ltd.
- Jochem Herrmann  
  Adimec Advanced Image Systems B.V.
- Tjeerd Veenstra  
  Adimec Advanced Image Systems B.V.
- Frederik Voncken  
  Adimec Advanced Image Systems B.V.
- Steve Mott  
  Components Express Inc.
- Bram Devuyst  
  EqcoLogic N.V.

C.3.2  The Japanese Translators of the Original CoaXPress Specification

The following people contributed to the translation of the original CoaXPress specification into Japanese:

- Masahide Matsubara  
  AVAL DATA Corporation
- Masaru Naganuma  
  Edec Linsey System Co., Ltd.
- Noriaki Tadokoro  
  Toshiba Teli Corporation
C.3.3  JIIA CoaXPress WG Members for v1.0

Leader of CoaXPress Working Group:

- Tadashi Miyazaki  AVAL DATA Corporation

Vice Leaders of CoaXPress Working Group:

- Masahito Watanabe  Adimec Electronic Imaging K.K.
- Hiroshi Fukui  CIS corporation
- Tsuneharu Iizuka  DAITO DENSO CO., LTD.

Leaders of Task Forces:

- Masahide Matsubara  AVAL DATA Corporation
- Hiroshi Fukui  CIS Corporation
- Tsuneharu Iizuka  DAITO DENSO CO., LTD.

Other Members of CoaXPress Working Group:

- Nobutaka Arai  Agilent Technologies, Inc.
- Yukio Komuro  Agilent Technologies, Inc.
- Tsuyoshi Ishii  Agilent Technologies, Inc.
- Katsumi Sakamoto  AnaFocus
- Hajime Unno  AVAL DATA Corporation
- Yuji Sasao  Basler Japan
- Takeshi Okuda  DAITO DENSO CO., LTD.
- Masaru Naganuma  Edec Linsey System Co., Ltd.
- Eiji Nishiyama  Fast Corporation
- Masahiro Kobayashi  Fast Corporation
- Soichiro Yamamoto  Fast Corporation
- Masaki Horikawa  GRAPHIN Co., Ltd.
- Takashi Kawakami  GRAPHIN Co., Ltd.
- Sadafumi Torii  Hamamatsu Photonics K.K.
- Hiroyuki Matsumoto  Hirakawa Hewtech corp.
- Eiji Ebihara  Hirakawa Hewtech corp.
- Susumu Nakagawa  Hirose Electric Co., Ltd.
- Hiroyuki Nishikawa  Hitachi Kokusai Electric Inc.
- Kohei Kageshita  Hitachi Kokusai Electric Inc.
- Atsunobu Sugimoto  Honda Tsushin Kogyo Co., Ltd.
- Kazuo Kasai  Honda Tsushin Kogyo Co., Ltd.
- Tsuyoshi Watanabe  JAI Ltd., Japan
- Mitsunari Kato  Micro Technica Co., Ltd.
- Yuki Hamamoto  Micro Technica Co., Ltd.
- Fumio Nagumo  Nagumo Professional Engineer Office
- Motofumi Kawanishi  National Semiconductor Japan Ltd.
- Yoshiki Yanagisawa  Nippon Electro-Sensory Devices Corporation
- Shinichi Masuyama  Nissei Electric Co., Ltd.
- Keiji Munekusa  Oki Electric Cable Co., Ltd.
- Hirokuni Kobayashi  Oki Electric Cable Co., Ltd.
- Hiroshi Yano  Omron Corporation
- Osamu Aimo  Sensor Technology Co., Ltd.
C.3.4 CoaXPress Liaison Members for v1.0
The following list is the CoaXPress Liaison members from the AIA and EMVA who contributed to the v1.0 standard:

- Chris Beynon  
  Active Silicon Ltd.
- Frederik Voncken  
  Adimec Advanced Image Systems B.V.
- Paul Kozik  
  Allied Vision Technologies GmbH
- Fritz Dierks  
  Basler AG
- Ray Berst  
  Components Express, Inc.
- John Egri  
  Imperx Inc
- Josh Morris  
  Nortech Systems’ Intercon 1
- Ron Folkeringa  
  Nortech Systems’ Intercon 1
- Michael Chee  
  Matrox Electronic Systems Ltd.
- Peter Schwider  
  Photonfocus AG
- Werner Feith  
  Sensor to Image GmbH.
- Dr. Holger Singpiel  
  Silicon Software GmbH
- Manuel Ott  
  Slectrix GmbH
- Rupert Stelz  
  STEMMER IMAGING GmbH

C.3.5 Contributors for v1.1
The following person was responsible for the production of the v1.1 standard:

- Chris Beynon  
  Active Silicon Ltd.

The following list is the people who have directly contributed to the v1.1 standard:

- Jean-Philippe Arnaud  
  Active Silicon Ltd.
- Chris Beynon  
  Active Silicon Ltd.
- Frederik Voncken  
  Adimec Advanced Image Systems B.V.
- Masahide Matsubara  
  AVAL DATA Corporation
- Dan Frisch  
  Bitflow, Inc.
- Reynold Dodson  
  Bitflow, Inc.
Additionally, the following people have contributed to the standardization process for v1.1:

- Masahito Watanabe, Adimec Electronic Imaging K.K.
- Tadashi Miyazaki, AVAL DATA Corporation
- Hiroshi Fukui, CIS Corporation
- Yoshinori Arai, Fujikura Ltd
- Hiroyuki Matsumoto, Hirakawa Hewtech corp.
- Tsuyoshi Watanabe, JAI Ltd., Japan
- Andreas Ertl, Miktrotron GmbH
- Keiji Munezuka, Oki Electric Cable Co., Ltd.
- Werner Feith, Sensor to Image GmbH
- Holger Singpiel, Silicon Software GmbH
- Kazuya Kudou, Toshiba Teli Corporation

### C.3.6 Contributors for v1.1.1

The v1.1.1 standard was edited by Chris Beynon of Active Silicon Ltd based on feedback from the CoaXPress community.
This page intentionally blank
The standards which the Japan Industrial Imaging Association publishes are enacted whether it infringes or not, on industrial property such as patents and new utility designs and so on.

一般社団法人日本インダストリアルイメージング協会が発行している規格類は、工業所有権（特許、実用新案など）に関する抵触の有無に関係なく制定されています。

The Japan Industrial Imaging Association is not responsible for any industrial property rights related to the contents of this standard.

一般社団法人日本インダストリアルイメージング協会は、この規格類の内容に関する工業所有権に対して、一切の責任を負いません。

JIIA CXP-001-2015

Published on Feb 24, 2016
2016 年 02 月 24 日発行

Published by
Japan Industrial Imaging Association

発行
一般社団法人 日本インダストリアルイメージング協会
2-21-27, Hyakunin-cho, Shinjyuku-ku, Tokyo 169-0073 Japan
TEL/FAX: +81-3-3361-6880

〒169-0073 東京都新宿区百人町 2-21-27
TEL/FAX: 03-3361-6880

All rights reserved by JIIA
禁無断転載

Information contained in this standard is subject to the Copyright Act of Japan, and the copyright protection controlled by international convention. When reprinting whole or a part of this standard, the permission from the publisher (JIIA) shall be obtained, except when it is for private use and or when it is met the Copyright Act.

この規格類に掲載の情報は、日本国の著作権法、及び国際条約による著作権保護の対象となっております。この規格類の内容について、私的使用、または引用等著作権法上認められた行為を除き、規格類の全部、または一部を転載しようとする場合は、発行者の許可をどうぞ。