

Japan Industrial Imaging Association Standard JIIA CP-001-2015 /
1394 Trade Association Specification TS2015001



IIDC2 Digital Camera Control Specification
Ver.1.1.0
May 19th, 2015

Sponsored by:

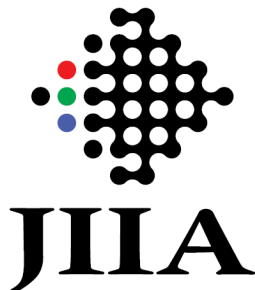
Japan Industrial Imaging Association (JIIA) / 1394 Trade Association (1394TA)

Accepted for publication by

JIIA and 1394TA Board of Directors.

Abstract

The purpose of this document is to act as a design guide for imaging device / host makers that wish to use a digital interface as the device-to-host interconnect. Adherence to the design specifications contained herein do not guarantee, but will promote interoperability for this class of device. The device registers, fields within those registers, video formats, modes of operation and controls for each are specified. Area has been left for growth. To make application for additional specification, contact the Japan Industrial Imaging Association IIDC2 Working Group or the 1394 Trade Association Industrial and Instrumentation Working Group. IIDC2 is designed for many kinds of digital interfaces.





Japan Industrial Imaging Association
The Standardization Committee –
IIDC2 Working Group

This document is offered “as is” in this current version.

JIIA, its members, subsidiary companies of its member nor the affiliates of the its members, do not guarantee any impact on trade, fitness for specific purpose, nor infringement of property rights.

JIIA, its members, subsidiary companies of its member nor the affiliates of the its members, do not take any responsibility for damage originating from the use of this text, or unavailability of this text (not limited to the passive damage and other derivative, incidental damage.) under applicable law.

This counts also if the damage is related to JIIA, its members, subsidiary companies of its member nor the affiliates of the its members.

JIIA, its members, subsidiary companies of its member nor the affiliates of the its members do not take the responsibility for defending not to compensate for potential disputes between JIIA and third party which may originate from intellectual property in this text.

The standard 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.

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

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.

Japan Industrial Imaging Association
2-21-27, Hyakunin-cho,
Shinjuku, Tokyo 169-0073, Japan

Copyright © 2015 by JIIA, 1394 TA
All rights reserved.

1394 Trade Association Specification



1394 Trade Association Specifications are developed within Working Groups of the 1394 Trade Association, a non-profit industry association devoted to the promotion of and growth of the market for IEEE 1394-compliant products. Participants in Working Groups serve voluntarily and without compensation from the Trade Association. Most participants represent member organizations of the 1394 Trade Association. The specifications developed within the working groups represent a consensus of the expertise represented by the participants.

Use of a 1394 Trade Association Specification is wholly voluntary. The existence of a 1394 Trade Association Specification is not meant to imply that there are not other ways to produce, test, measure, purchase, market or provide other goods and services related to the scope of the 1394 Trade Association Specification. Furthermore, the viewpoint expressed at the time a specification is accepted and issued is subject to change brought about through developments in the state of the art and comments received from users of the specification. Users are cautioned to check to determine that they have the latest revision of any 1394 Trade Association Specification.

Comments for revision of 1394 Trade Association Specifications are welcome from any interested party, regardless of membership affiliation with the 1394 Trade Association. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally, questions may arise about the meaning of specifications in relationship to specific applications. When the need for interpretations is brought to the attention of the 1394 Trade Association, the Association will initiate action to prepare appropriate responses.

Comments on specifications and requests for interpretations should be addressed to:

Editor, 1394 Trade Association
23117 39th Ave SE
Bothell, WA 98021 USA

1394 Trade Association Specifications are adopted by the 1394 Trade Association without regard to patents which may exist on articles, materials or processes or to other proprietary intellectual property which may exist within a specification. Adoption of a specification by the 1394 Trade Association does not assume any liability to any patent owner or any obligation whatsoever to those parties who rely on the specification documents. Readers of this document are advised to make an independent determination regarding the existence of intellectual property rights, which may be infringed by conformance to this specification.

Contents

| | |
|--|----|
| Foreword..... | 1 |
| Version history | 2 |
| 1 Scope and purpose | 3 |
| 1.1 Scope | 3 |
| 1.2 Purpose | 3 |
| 2 Definitions and notation | 4 |
| 2.1 Definitions | 4 |
| 2.1.1 Requirements Terminology..... | 4 |
| 2.1.2 Glossary | 5 |
| 2.1.3 Abbreviations..... | 5 |
| 2.2 Notation | 5 |
| 2.2.1 Numeric values | 5 |
| 2.2.2 Bit, byte and quadlet ordering..... | 5 |
| 3 Digital device control register | 7 |
| 3.1 IIDC2Entry address | 7 |
| 3.2 General structure..... | 7 |
| 3.3 IIDC2Entry (offset list of CategoryBlock) | 8 |
| 3.3.1 Keyword | 9 |
| 3.3.2 Version..... | 9 |
| 3.3.3 NumberOfCategoryBlocks | 9 |
| 3.3.4 OffsetForXmlManifestTable..... | 9 |
| 3.3.5 OffsetForCategoryBlock0 ~ 31..... | 9 |
| 3.4 Structure of CategoryBlock | 10 |
| 3.4.1 CategoryBlock List..... | 10 |
| 3.4.2 Header of CategoryBlock | 11 |
| 3.5 Standard structure of FeatureCSR | 12 |
| 3.5.1 Inquiry | 12 |
| 3.5.2 OffsetForExpanded | 14 |
| 3.5.3 Control..... | 15 |
| 3.5.3.1 NoSpecify | 15 |
| 3.5.3.2 Manual | 16 |
| 3.5.3.3 Auto | 16 |
| 3.5.3.4 AutoOnce | 17 |
| 3.5.3.5 Default | 17 |
| 3.5.4 ValueRegisterArea (Common register type) | 17 |
| 3.5.4.1 Integer..... | 17 |
| 3.5.4.2 PlainInteger..... | 19 |
| 3.5.4.3 Float | 20 |
| 3.5.4.4 Enumeration..... | 21 |
| 3.5.4.5 BulkBoolean | 21 |
| 3.5.4.6 Rectangle | 22 |
| 3.5.4.7 Array of Register..... | 24 |
| 3.5.4.8 String..... | 26 |
| 3.5.5 Multi-Byte accessing | 26 |
| 3.6 XmlManifestTable | 27 |
| 3.6.1 XmlManifestSize | 27 |
| 3.6.2 XmlManifestSelector | 27 |
| 3.6.3 XmlVersion | 27 |
| 3.6.4 XmlSchemaVersion | 27 |
| 3.6.5 XmlUrlAddress..... | 28 |
| 3.6.5.1 URL Format - Non-Volatile Memory..... | 28 |
| 3.6.5.2 URL Format – Vendor website | 29 |
| 4 Device Control Register | 30 |
| 4.1 CategoryBlock0 (DeviceControl) | 31 |
| 4.1.1 DeviceReset | 33 |
| 4.1.2 DevicePower..... | 34 |
| 4.1.3 DeviceVendorName..... | 34 |

| | |
|---|----|
| 4.1.4 DeviceModelName | 34 |
| 4.1.5 DeviceManufacturerInfo | 34 |
| 4.1.6 DeviceVersion | 34 |
| 4.1.7 DeviceFirmwareVersion | 34 |
| 4.1.8 DeviceID | 34 |
| 4.1.9 DeviceUserID | 35 |
| 4.1.10 LEDIndicatorLuminance | 35 |
| 4.2 CategoryBlock1 (TransportLayerControl) | 36 |
| 4.3 CategoryBlock2 (ImageFormatControl) | 37 |
| 4.3.1 ImageFormatSelector | 44 |
| 4.3.2 ApplyImageFormat | 45 |
| 4.3.3 ImageSize | 46 |
| 4.3.4 PixelCoding / PixelSize | 46 |
| 4.3.5 PixelEndian | 47 |
| 4.3.6 BinningHorizontal / BinningVertical | 47 |
| 4.3.7 DecimationHorizontal / DecimationVertical | 47 |
| 4.3.8 ReverseX / ReverseY | 47 |
| 4.3.9 TestPattern | 47 |
| 4.4 CategoryBlock3 (AcquisitionControl) | 48 |
| 4.4.1 AcquisitionCommand | 50 |
| 4.4.2 AcquisitionFrameCount | 50 |
| 4.4.3 ImageBufferMode | 50 |
| 4.4.4 ImageBufferFrameCount | 52 |
| 4.4.5 AcquisitionFrameRate | 52 |
| 4.4.6 AcquisitionFrameInterval | 52 |
| 4.5 CategoryBlock4 (LuminanceControl) | 53 |
| 4.5.1 ExposureTime | 54 |
| 4.5.2 BlackLevel | 54 |
| 4.5.3 Gain | 54 |
| 4.5.4 Gamma | 54 |
| 4.5.5 Sharpness | 55 |
| 4.5.6 ALCLLevel | 55 |
| 4.6 CategoryBlock5 (ChromaControl) | 56 |
| 4.6.1 Hue | 57 |
| 4.6.2 Saturation | 57 |
| 4.6.3 WhiteBalanceR / WhiteBalanceB | 57 |
| 4.6.4 WhiteBalanceU / WhiteBalanceV | 57 |
| 4.6.5 ColorTemperature | 58 |
| 4.7 CategoryBlock6 (LUTControl) | 59 |
| 4.7.1 LUTEnable | 60 |
| 4.7.2 LUTBankSelector | 60 |
| 4.7.3 LUTValueAll | 61 |
| 4.8 CategoryBlock7 (TriggerControl) | 62 |
| 4.8.1 TriggerMode | 66 |
| 4.8.2 TriggerSequence | 66 |
| 4.8.3 TriggerSource | 70 |
| 4.8.4 TriggerAdditionalParameter | 70 |
| 4.8.5 TriggerDelay | 70 |
| 4.8.6 SoftwareTrigger | 70 |
| 4.9 CategoryBlock8 (UserSetControl) | 71 |
| 4.9.1 StandardFormat / StandardFrameRate | 74 |
| 4.9.2 UserSetSelector / UserSetCommand | 75 |
| 4.9.3 UserSetDefault | 76 |
| 4.10 CategoryBlock9 (DigitalIOControl) | 77 |
| 4.10.1 IOLineModeAll | 82 |
| 4.10.2 IOLineInverterAll | 82 |
| 4.10.3 IOLineStatusAll | 82 |
| 4.10.4 UserOutputValueAll | 82 |
| 4.10.5 IOLineSelector | 82 |
| 4.10.6 IOLineSource | 83 |

| | |
|--|----|
| 4.11 CategoryBlock10 (CounterAndTimerControl) | 84 |
| 4.11.1 TimerSelector | 86 |
| 4.11.2 TimerDelay / TimerDuration | 86 |
| 4.11.3 TimerTriggerSource | 86 |
| 4.12 CategoryBlock11 (EventControl) | 87 |
| 4.12.1 EventNotificationOfFame / EventNotificationOfExposure | 89 |
| 4.13 CategoryBlock24~31 (VendorUniqueControl) | 90 |
| 5 FeatureCSR Chaining | 92 |
| 5.1 Use case of common logic | 92 |
| 5.1.1 Value | 92 |
| 5.1.2 Control | 92 |
| 5.2 Use case of independent logics | 93 |
| 5.2.1 Value | 93 |
| 5.2.2 Control | 93 |
| Document history | 99 |

Tables

| | |
|--|----|
| Table 1 – Requirements Terminology | 4 |
| Table 2 – Suffix Terminology | 4 |
| Table 3 –IIDC2Entry | 8 |
| Table 4 – BasicCategoryBlock list | 10 |
| Table 5 – Structure of CategoryBlock | 10 |
| Table 6 – Header of CategoryBlock | 11 |
| Table 7 – Structure of FeatureCSR | 12 |
| Table 8 – Inquiry | 12 |
| Table 9 – Truth table of Inquiry | 14 |
| Table 10 – OffsetForExpanded | 14 |
| Table 11 – Control | 15 |
| Table 12 – Integer32 (ValueType = 0x30) | 17 |
| Table 13 – Integer64 (ValueType = 0x40) | 18 |
| Table 14 – PlainInteger8 (ValueType = 0x11) | 19 |
| Table 15 – PlainInteger32 (ValueType = 0x31) | 19 |
| Table 16 – PlainInteger64 (ValueType = 0x41) | 19 |
| Table 17 – Float32 (ValueType = 0x32) | 20 |
| Table 18 – Float64 (ValueType = 0x42) | 20 |
| Table 19 – Enumeration (ValueType = 0x03) | 21 |
| Table 20 – BulkBoolean32 (ValueType = 0x34) | 21 |
| Table 21 – BulkBoolean64 (ValueType = 0x44) | 22 |
| Table 22 – Rectangle32 (ValueType = 0x35) | 22 |
| Table 23 – Quadlet of ArrayInformationArea | 24 |
| Table 24 – Example of ArrayOfInteger32 with 2 dimension (ValueType = 0xB0) | 25 |
| Table 25 – Example of String with 16 characters (ValueType = 0x91) | 26 |
| Table 26 – XmlManifestTable | 27 |
| Table 27 – URL format – Non-volatile memory | 28 |
| Table 28 – URL format – Vendor website | 29 |
| Table 29 –CategoryBlock0 (DeviceControl) | 31 |
| Table 30 – CategoryBlock2 (ImageFormatControl) | 37 |
| Table 31 – Pixel Coding, PixelSize and corresponding PixelFormat | 46 |
| Table 32 – CategoryBlock3 (AcquisitionControl) | 48 |
| Table 33 – CategoryBlock4 (LuminanceControl) | 53 |
| Table 34 – CategoryBlock5 (ChromaControl) | 56 |
| Table 35 – CategoryBlock6 (LUTControl) | 59 |
| Table 36 – LUTValueAll in ExpandedCategoryBlock (chaining CSR) | 61 |
| Table 37 – CategoryBlock7 (TriggerControl) | 62 |
| Table 38 – CategoryBlock8 (UserSetControl) | 71 |
| Table 39 –Relationship between ImageSize and type of StandardFormat | 75 |
| Table 40 – CategoryBlock9 (DigitalIOControl) | 77 |

Table 41 – CategoryBlock10 (CounterAndTimerControl) 84
 Table 42 –CategoryBlock11 (EventControl) 87
 Table 43 – EventID list 88
 Table 44 – CategoryBlock24~31 (VendorUniqueControl) 90
 Table 45 – Recommended FeatureCSRs..... 94
 Table 46 – OffsetForExpanded in IIDC2 Ver.1.0.0..... 98

Figures

Figure 1 – Bit ordering within a quadlet 5
 Figure 2 – Byte ordering within a quadlet at Big Endian 6
 Figure 3 – Byte ordering within a quadlet at Little Endian 6
 Figure 4 –Quadlet ordering within a several quadlets area at Big Endian 6
 Figure 5 –Quadlet ordering within a several quadlets area at Little Endian 6
 Figure 6 – General Structure..... 7
 Figure 7 – Elements of Rectangle type 23
 Figure 8 – Array of register 24
 Figure 9 – ImageFormatSelector 44
 Figure 10 – State machine of ApplyImageFormat 45
 Figure 11 – Block diagram of AcquisitionCommand and ImageBufferMode 51
 Figure 12 – Example timing of image buffer control 51
 Figure 13 – Trigger Sequence0..... 66
 Figure 14 – Trigger Sequence1 66
 Figure 15 – Trigger Sequence2..... 67
 Figure 16 – Trigger Sequence3..... 67
 Figure 17 – Trigger Sequence4..... 68
 Figure 18 – Trigger Sequence5..... 68
 Figure 19 – Trigger Sequence6..... 69
 Figure 20 – StandardFormat / StandardFrameRate..... 74
 Figure 21 – UserSetSelector / UserSetCommand 75
 Figure 22 – Block diagram of DigitalIOControl..... 81
 Figure 23 – Output signal definition..... 83
 Figure 24 – Definition of TimerActive signal..... 86
 Figure 25 – FeatureCSR longer than 8 quadlets 91
 Figure 26 – Gain amplifier which has Integer32 / Float32 register 92
 Figure 27 –2 gain amplifier logics 93
 Figure 28 – Example of multiple-ROI 95
 Figure 29 – Structure of multiple-ROI..... 96
 Figure 30 – CSR Chaining of IIDC2 Ver.1.0.0 97

Annexes

Annex A Recommended FeatureCSRs of camera devices..... 94
 Annex B Supporting multiple-ROI..... 95
 B.1 Overview..... 95
 B.2 BasicCSR..... 96
 B.3 ROIEnable 96
 B.4 ROIImageSize..... 96
 Annex C Keeping compatibility for previous version of IIDC2..... 97
 C.1 CSR Chaining mechanism (for host) 97

Requirements

[R-1cd]..... 9
 [R-2cd]..... 9
 [R-3cd]..... 9
 [CR-4cd]..... 9
 [CR-5cd]..... 9
 [CR-6cd]..... 9
 [CR-7ch]..... 9
 [CR-8cd]..... 14
 [R-9cd]..... 15
 [O-10cd] 15
 [CR-11cd] 15
 [R-12cd]..... 15
 [O-13cd] 16
 [O-14ch] 16
 [O-15cd] 16
 [CR-16cd]..... 16
 [O-17cd] 16
 [CR-18cd] 16
 [R-19cd]..... 16
 [R-20cd]..... 17
 [O-21cd] 17
 [CR-22cd]..... 17
 [R-23cd]..... 17
 [R-24cd]..... 17
 [R-25cd]..... 17
 [R-26c]..... 18
 [CR-27c]..... 18
 [CR-28cd] 18
 [CR-29cd] 19
 [R-30cd]..... 20
 [R-31cd]..... 20
 [R-32cd]..... 21
 [R-33cd]..... 22
 [R-34c]..... 23
 [R-35cd]..... 26
 [CR-36c] 27
 [R-37c]..... 27
 [R-38c]..... 28
 [CR-39c]..... 28
 [CO-40c]..... 28
 [CO-41c]..... 29
 [R-42c]..... 30
 [CR-43ch]..... 30

| | |
|-----------------|----|
| [CR-44cd] | 33 |
| [CR-45c] | 33 |
| [R-46cd] | 33 |
| [R-47cd] | 33 |
| [R-48ch] | 33 |
| [O-49cd] | 34 |
| [CR-50cd] | 35 |
| [R-51cd] | 44 |
| [R-52ch] | 45 |
| [CR-53cd] | 45 |
| [R-54cd] | 45 |
| [R-55cd] | 45 |
| [R-56ch] | 46 |
| [R-57cd] | 52 |
| [O-58ch] | 52 |
| [CO-59cd] | 52 |
| [R-60cd] | 52 |
| [O-61ch] | 52 |
| [O-62cd] | 52 |
| [CR-63cd] | 52 |
| [CR-64cd] | 57 |
| [CR-65cd] | 57 |
| [CR-66c] | 61 |
| [CCR-67c] | 67 |
| [CR-68cd] | 67 |
| [CR-69cd] | 68 |
| [CR-70cd] | 68 |
| [CR-71cd] | 69 |
| [R-72cd] | 70 |
| [R-73cd] | 74 |
| [R-74cd] | 82 |
| [R-75cd] | 82 |
| [CR-76ch] | 83 |
| [CR-77ch] | 83 |
| [CR-78c] | 88 |
| [R-79c] | 91 |
| [CR-80c] | 91 |
| [CR-81cd] | 91 |
| [R-82c] | 91 |
| [R-83cd] | 91 |
| [O-84ch] | 92 |
| [CR-85ch] | 92 |
| [O-86ch] | 92 |
| [CO-87cd] | 96 |
| [CO-88ch] | 98 |

Foreword

(This foreword is not part of JIIA Standard JIIA CP-001-2015 / 1394TA Specification TS2015001)

This specification defines IIDC2 Digital Camera Control Specification Ver.1.1.0 for industrial use. This document describes control and status registers definition and its control procedures.

The 1st version of IIDC (before IIDC Ver.1.32) attributes to 1394 Trade Association (hereinafter called 1394TA). The 2nd version of IIDC (IIDC2) attributes to Japan Industrial Imaging Association (hereinafter called JIIA), and be copyrighted by JIIA and 1394TA.

This specification was accepted by the Board of Directors of the JIIA and 1394TA. Board of Directors acceptance of this specification does not necessarily imply that all board members voted for acceptance. At the time it accepted this specification, the JIIA. Board of Directors had the following members:

BOD of JIIA in July, 2015

| <i>Post</i> | <i>Organization Represented</i> | <i>Name of Representative</i> |
|------------------|-----------------------------------|-------------------------------|
| Chair | Symco..... | Sachio Kiura |
| Vice-Chair..... | MICRO-TECHNICA | Setsuko Iwata |
| Vice-Chair..... | CIS | Hiroshi Fukui |
| Vice-Chair | Adimec Electronic Imaging | Masahito Watanabe |
| Secretary..... | Advanced Communication Media..... | Norichika Yui |
| BOD..... | CIC | Yutaka Takaoka |
| BOD..... | Hitachi Kokusai Electric | Akimi Tsukui |
| BOD..... | Imac..... | Hiroshi Totsuka |
| BOD..... | Kirin Techno System | Iwao Nishimura |
| BOD..... | Optart | Kazuo Fujiyama |
| BOD..... | ADSTEC | Hiroshi Yako |
| BOD..... | Toshiba Teli | Masanori Yasuda |

BOD of 1394TA

| <i>Post</i> | <i>Organization Represented</i> | <i>Name of Representative</i> |
|-----------------------|---------------------------------|-------------------------------|
| Chair | Lugeo | Richard Mourn |
| Secretary | Intel | Dave Thompson |
| Financial Officer.... | IPRA..... | Richard Davies |

The IIDC2 Working Group in JIIA and the Industrial and Instrumentation Working Group in 1394TA, which developed and reviewed this specification, had the following members:

BOD of IIDC2 WG in JIIA

| | | |
|---------------------|---------------------|------------|
| Sadafumi Torii | Hamamatsu Photonics | Chair |
| Yoshitsugu Nakasone | Toshiba Teli | Vice-chair |
| Masaki Horikawa | GRAPHIN | Vice-chair |
| Koji Tsuchiya | TechnoScope | Vice-chair |

BOD of Industrial and Instrumentation WG in 1394TA

| | | |
|---------------|---------------------|--------------------------------|
| Dave Thompson | Intel | Chair |
| Rod Barman | Point Grey Research | Chair of Digital Camera sub-WG |

Other contributed members

| | |
|------------------|---------------------|
| Junji Kishi | Toshiba Teli |
| Yasutoshi Onishi | Hamamatsu Photonics |
| Tokuhiro Koike | Hamamatsu Photonics |
| Takayuki Inoue | Hamamatsu Photonics |
| Akira Mizutani | DITECT |
| Masato Tsujie | Sony |

Other members

| | |
|--------------|------------------------|
| Hiroki Nagao | NEC Engineering |
| Max Bassler | Interactive Technology |

Version history

Version 1.0.0 (January 13th, 2012) first version

Version 1.1.0 (May 19th, 2015)

IIDC2 Digital Camera Control Specification Ver.1.1.0

1 Scope and purpose

1.1 Scope

This document specifies the IIDC2 Digital Camera Control Specification Ver.1.1.0 for industrial use. This document describes control and status register definitions and control procedures.

1.2 Purpose

The purpose of this specification is to describe IIDC2 Digital Camera Control Specification Ver.1.1.0 for Digital Camera and industrial peripheral devices.

This specification defines a set of addressed register. This specification was designed to be easily applied to different digital interfaces (such as IEEE1394, CoaXPress, Camera Link, USB3 Vision and etc...) and adopted by their upper level protocols.

2 Definitions and notation

2.1 Definitions

2.1.1 Requirements Terminology

This standard uses The conventions shown as follows to list requirements:

| Term | Description | Representation |
|-------------------------|--|-------------------|
| Absolute Requirement | Feature that SHALL be supported by the product. It is mandatory to support the feature to ensure interoperability. | [R-<sn><suffix>] |
| Conditional Requirement | Feature that SHALL be supported if another feature is present. It is mandatory to support the feature when another feature is supported. | [CR-<sn><suffix>] |
| Absolute Objective | Feature that SHOULD be supported by the product. It is recommended, but not essential. | [O-<sn><suffix>] |
| Conditional Objective | Feature that SHOULD be supported if another feature is present. It is recommended, but not essential. | [CO-<sn><suffix>] |

Table 1 – Requirements Terminology

Each requirement is represented by an unique number in brackets. Each number is composed by 3 elements:

1st element: Requirement Type: Absolute Requirement, Conditional Requirement, Absolute Objective, Conditional Objective

2nd element: Sequence Number: Unique number identifying the requirement or objective. The sequence numbers are attributed sequentially as new requirements and objectives are added to the standard.

3rd element: Suffix: Identifies if requirement or objective is applicable to hosts or devices. These terminologies are shown as follows.

| Meaning | Description | Suffix |
|----------------|---|--------|
| Control | Represents a control-related requirement or objective applicable to both devices and hosts. | c |
| Control Host | Represents a control-related requirement or objective exclusive to hosts. | ch |
| Control Device | Represents a control-related requirement or objective exclusive to devices. | cd |

Table 2 – Suffix Terminology

2.1.2 Glossary

The following terms are used in this specification:

1 quadlet: four bytes of data

2 AbsoluteValue: normalized value, its unit and scale are defined by specification. It is a virtual register (not implemented in the device). Host can get it with calculation.

2.1.3 Abbreviations

The following are abbreviations that are used in this specification:

CSR Control and status register

As exemplified by CSR, abbreviations may cite a bibliographic reference.

2.2 Notation

2.2.1 Numeric values

Decimal and hexadecimal are used within this specification. By editorial convention, decimal numbers are most frequently used to represent quantities or counts. Addresses are uniformly represented by hexadecimal numbers. Hexadecimal numbers are also used when the value represented has an underlying structure that is more apparent in a hexadecimal format than in a decimal format.

Decimal numbers are represented by Arabic numerals without subscripts or by their English names. Hexadecimal numbers are represented by digits from the character set 0 – 9 and A – F with prefix code of 0x (C-language style). For the sake of legibility hexadecimal numbers are separated into groups of four digits separated by spaces.

As an example, 42 and 0x2A both represent the same numeric value.

2.2.2 Bit, byte and quadlet ordering

This specification uses the facilities of interface and therefore uses the ordering conventions of interface in the representation of data structures. In order to promote interoperability with memory buses that may have different ordering conventions, this specification defines the order and significance of bits within quadlet, bytes within quadlets and quadlets within several quadlets area in terms of their relative position and not their physically addressed position.

At bit ordering within a quadlet, the most significant bit (msb) is left bound bit, and least significant bit (lsb) is right bound bit. Other else bit fields are same as it.

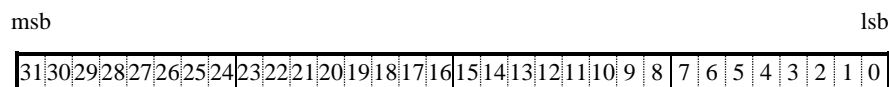


Figure 1 – Bit ordering within a quadlet

There are two types of byte ordering within a quadlet: One is Big Endian, the other is Little Endian. Which ordering to use is depend on the specification of interface.

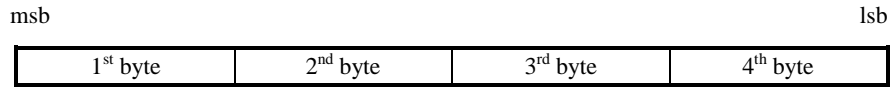


Figure 2 – Byte ordering within a quadlet at Big Endian

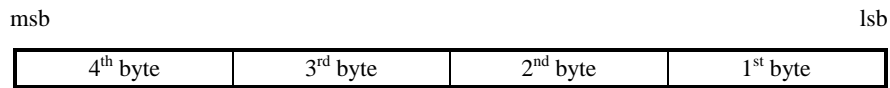


Figure 3 – Byte ordering within a quadlet at Little Endian

Similarly, there are two types of quadlet ordering within a several quadlets area.

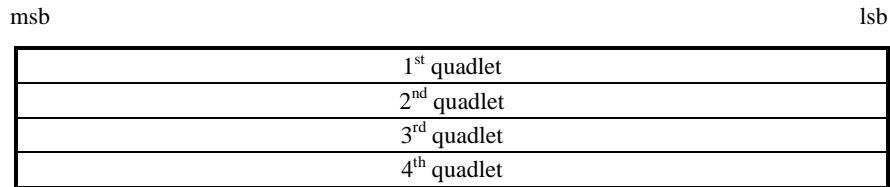


Figure 4 –Quadlet ordering within a several quadlets area at Big Endian

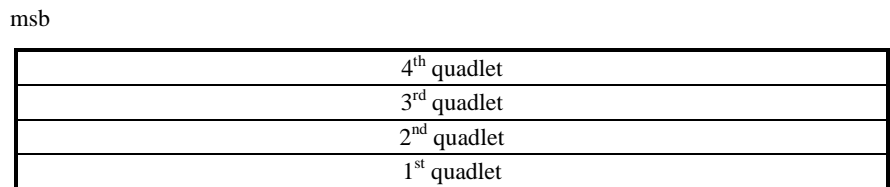


Figure 5 –Quadlet ordering within a several quadlets area at Little Endian

3 Digital device control register

3.1 IIDC2Entry address

IIDC2 register space starts from IIDC2Entry

3.2 General structure

IIDC2Entry has the address offset list of CategoryBlocks. Each CategoryBlock has some FeatureCSRs which are categorized. Every feature has one bunch of registers (FeatureCSR). CategoryBlocks contain BasicCSRs which provide the way to handle standard device functions. BasicCSRs may have one or several ExpandedCSRs with chaining mechanism. OffsetForExpanded in FeatureCSR points the address of chaining ExpandedCSR. Devices may use ExpandedCSRs for these proprietary specifications.

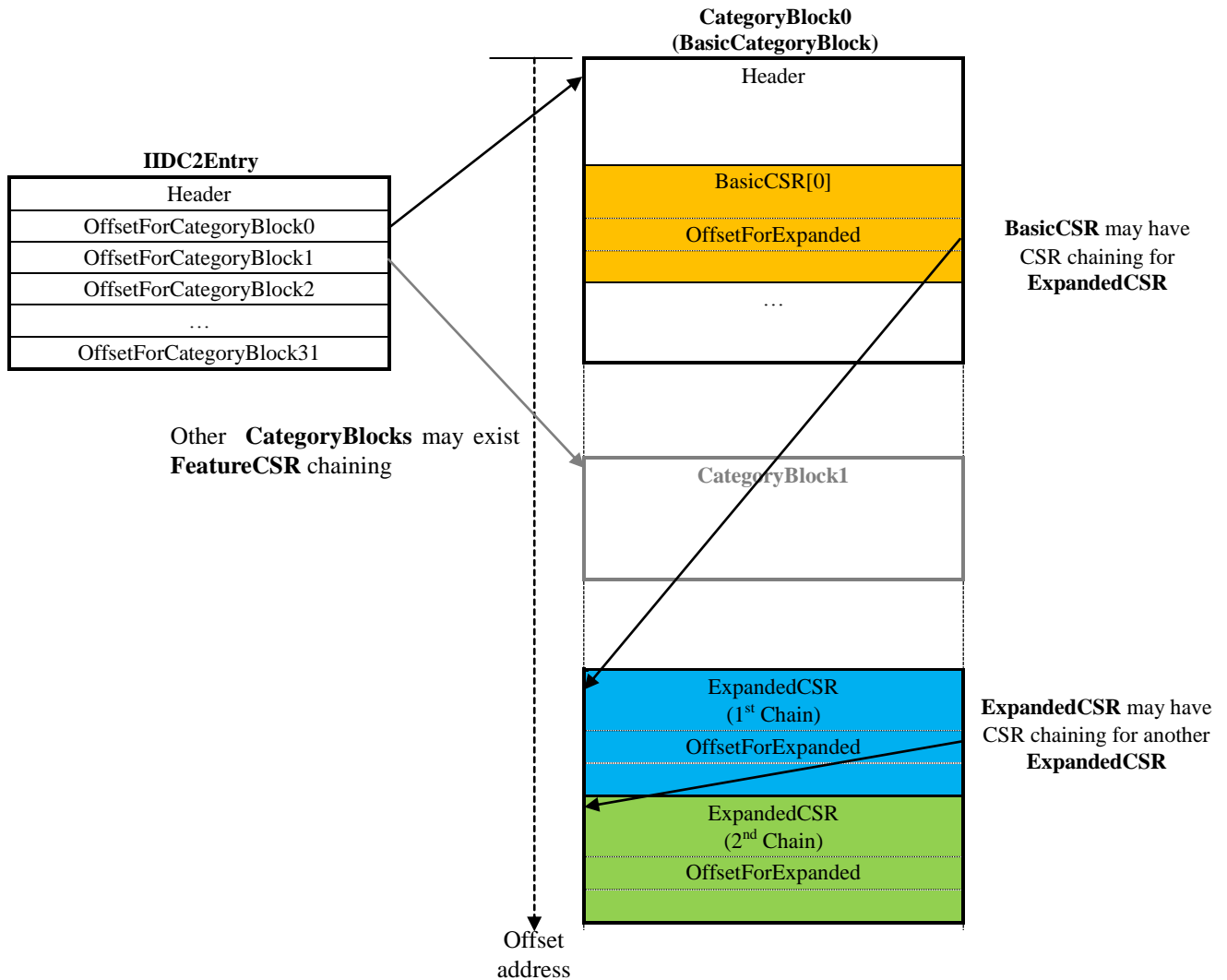


Figure 6 – General Structure

3.3 IIDC2Entry (offset list of CategoryBlock)

IIDC2Entry contains the list of offset address for **CategoryBlock**. Each feature has one bunch of control registers and status registers(**CSR**). All Features are categorized and located inside each **CategoryBlock**.

Device that doesn't implement the **CategoryBlock** which is not necessary may set the offset address to zeros.

| Offset | Name | Description | |
|-------------------|---------------------------|---------------------------------------|-------------------------------|
| +0x000 | Keyword | 0x4949 4443 ("IIDC") | |
| +0x004 | Version | [31..24] | reserved |
| | | [23..16] | Major version number |
| | | [15..8] | Minor version number |
| | | [7..0] | Sub-minor version number |
| +0x008 | NumberOfCategoryBlocks | [31..16] | reserved |
| | | [15..0] | Number of CategoryBlocks (=N) |
| +0x00C | - | reserved | |
| +0x010 | OffsetForXmlManifestTable | Offset address for XML manifest table | |
| +0x014~ +0x01C | - | reserved | |
| +0x020 | OffsetForCategoryBlock0 | Offset address for CategoryBlock0 | |
| +0x024 | OffsetForCategoryBlock1 | Offset address for CategoryBlock1 | |
| +0x028 | OffsetForCategoryBlock2 | Offset address for CategoryBlock2 | |
| +0x02C | OffsetForCategoryBlock3 | Offset address for CategoryBlock3 | |
| +0x030 | OffsetForCategoryBlock4 | Offset address for CategoryBlock4 | |
| +0x034 | OffsetForCategoryBlock5 | Offset address for CategoryBlock5 | |
| +0x038 | OffsetForCategoryBlock6 | Offset address for CategoryBlock6 | |
| +0x03C | OffsetForCategoryBlock7 | Offset address for CategoryBlock7 | |
| +0x040 | OffsetForCategoryBlock8 | Offset address for CategoryBlock8 | |
| +0x044 | OffsetForCategoryBlock9 | Offset address for CategoryBlock9 | |
| +0x048 | OffsetForCategoryBlock10 | Offset address for CategoryBlock10 | |
| +0x04C | OffsetForCategoryBlock11 | Offset address for CategoryBlock11 | |
| +0x050~ +0x07C | - | reserved | |
| +0x080 | OffsetForCategoryBlock24 | Offset address for CategoryBlock24 | |
| +0x084 | OffsetForCategoryBlock25 | Offset address for CategoryBlock25 | |
| +0x088 | OffsetForCategoryBlock26 | Offset address for CategoryBlock26 | |
| +0x08C | OffsetForCategoryBlock27 | Offset address for CategoryBlock27 | |
| +0x090 | OffsetForCategoryBlock28 | Offset address for CategoryBlock28 | |
| +0x094 | OffsetForCategoryBlock29 | Offset address for CategoryBlock29 | |
| +0x098 | OffsetForCategoryBlock30 | Offset address for CategoryBlock30 | |
| +0x09C | OffsetForCategoryBlock31 | Offset address for CategoryBlock31 | |

Table 3 –IIDC2Entry

3.3.1 Keyword

[R-1cd]

Device shall contain the string “IIDC” in **Keyword** field.

3.3.2 Version

Indicates the version of IIDC specification.

[R-2cd]

Device shall contain the quadlet 0x0001 0100 in **Version** field.

3.3.3 NumberOfCategoryBlocks

Indicates the number of **CategoryBlock** a device has. It includes reserved space.

[R-3cd]

Device shall have **NumberOfCategoryBlocks** field.

3.3.4 OffsetForXmlManifestTable

Indicates offset address of XML Manifest table which is shown at section 3.6 XmlManifestTable. Offset address is the relative address from the top of **IIDC2Entry**. XML Manifest Table starts at **IIDC2Entry** address + **OffsetForXmlManifestTable** value.

XML Manifest table in IIDC2 is used for only interfaces which have no XML Manifest in transport layer (e.g. IEEE1394).

[CR-4cd]

OffsetForXmlManifestTable shall be multiple of quadlet (4 Bytes).

[CR-5cd]

If the interface uses an XML Manifest in the transport layer (e.g. CoaXPress, USB3 Vision), this field shall be not-used.

3.3.5 OffsetForCategoryBlock0 ~ 31

Indicates the offset addresses of each CategoryBlock. Offset address is the relative address from the top of **IIDC2Entry**. **CategoryBlockN** (N=0~31) starts at **IIDC2Entry** address + **OffsetForCategoryBlockN** value.

[CR-6cd]

OffsetForCategoryBlock0~31 shall be multiple of quadlet (4 Bytes).

[CR-7ch]

If **OffsetForCategoryBlock0~31** is zero, host shall not refer corresponding **CategoryBlock**.

3.4 Structure of CategoryBlock

3.4.1 CategoryBlock List

The standard features (e.g. Exposure time, Gain and etc...) are categorized and placed in **BasicCategoryBlock**. Relationship between index of **CategoryBlock** and category name is described in the following table.

| Name | Description |
|--------------------------------------|------------------------|
| CategoryBlock0 | DeviceControl |
| CategoryBlock1 | TransportLayerControl |
| CategoryBlock2 | ImageFormatControl |
| CategoryBlock3 | AcquisitionControl |
| CategoryBlock4 | LuminanceControl |
| CategoryBlock5 | ChromaControl |
| CategoryBlock6 | LUTControl |
| CategoryBlock7 | TriggerControl |
| CategoryBlock8 | UserSetControl |
| CategoryBlock9 | DigitalIOControl |
| CategoryBlock10 | CounterAndTimerControl |
| CategoryBlock11 | EventControl |
| CategoryBlock24 ~ CategoryBlock31 | VendorUniqueControl |

Table 4 – BasicCategoryBlock list

CategoryBlock contains only **BasicCSRs**. All **BasicCSRs** are located to fixed addressing which is defined by **IIDC2**.

| Offset | Description |
|--------|-------------|
| +0x000 | Header |
| +0x020 | BasicCSR[0] |
| +0x040 | BasicCSR[1] |
| +0x060 | BasicCSR[2] |
| ... | ... |

Table 5 – Structure of CategoryBlock

3.4.2 Header of CategoryBlock

| Offset | Name | Bit | Description |
|------------------|--------|----------|--------------------------------|
| +0x000 | Header | [31..24] | CategoryBlockNumber |
| | | [23..0] | SizeOfCategoryBlock (by Bytes) |
| +0x004~ 0x01C | | Reserved | |

Table 6 – Header of CategoryBlock

CategoryBlockNumber is the index of CategoryBlock.

SizeOfCategoryBlock indicates the size of CategoryBlock (including header area) in Bytes.

3.5 Standard structure of FeatureCSR

| Offset | Name | 31-24 | | | | | | 23-16 | | | | | | | | 15-8 | | | | | | | | 7-0 | | | | | | | | | | |
|--------|---------------------|-------------|--------|--|--|--|--|-----------------|----------|----------|-----------|--|--|--|--|------------|--|--|--|--|--|--|--|-----|--|---------|--|--|--|------------|-------------|---------|-----------|--------------|
| +0x00 | Inquiry | Implemented | Active | | | | | UserSetLoadable | Writable | Readable | ValueType | | | | | | | | | | | | | | | | | | | DefaultInq | AutoOnceInq | AutoInq | ManualInq | NoSpecifyInq |
| | | FeatureInq | | | | | | | | | | | | | | ControlInq | | | | | | | | | | | | | | | | | | |
| +0x04 | OffsetFor Expanded | | | | | | | OffsetAddress | | | | | | | | | | | | | | | | | | | | | | | | | | |
| +0x08 | Control | | | | | | | | | | | | | | | | | | | | | | | | | Control | | | | | | | | |
| +0x0C~ | Value Register Area | Value | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Table 7 – Structure of FeatureCSR

3.5.1 Inquiry

The first quadlet represents basic information about the feature. All bits here are read-only.

The upper 8-bits are **FeatureInq**. It describes information of this **FeatureCSR**. The middle 8-bits are **ValueType**. It indicates the type of **ValueRegisterArea**. The lower 16-bits are **ControlInq**. It indicates the available controls in this **FeatureCSR**.

| Bit | Field | Description |
|------|-------------|---|
| [31] | Implemented | 1 : Implemented 0 : Not Implemented This field represents whether device has this feature. This field never changes. |
| [30] | Active | 1 : Active 0 : Inactive This field represents whether this feature is active currently. |

Table 8 – Inquiry

| Bit | Field | Description |
|----------|-----------------|---|
| [26] | UserSetLoadable | 1 : Value and Control are updated when UserSetControl = Load . 0 : Not to be updated This field represents whether this feature setting is to be loaded or not when UserSetControl = Load . Device may change this field according the current device status. |
| [25] | Writable | 1 : Value is writable 0 : Value is not writable This field represents whether Value is writable or not. |
| [24] | Readable | 1 : Value is readable 0 : Value is not readable This field represents whether Value is readable or not. |
| [23..16] | ValueType | Described in the section 3.5.4 ValueRegisterArea (Common register type) This field represents Value type. |
| [4] | DefaultInq | 1 : Default is available 0 : Default is not available This field represents whether Default control is available or not. |
| [3] | AutoOnceInq | 1 : AutoOnce is available 0 : AutoOnce is not available This field represents whether AutoOnce control is available or not. |
| [2] | AutoInq | 1 : Auto is available 0 : Auto is not available This field represents whether Auto control is available or not. |
| [1] | ManualInq | 1 : Manual is available 0 : Manual is not available This field represents whether Manual control is available or not. |
| [0] | NoSpecifyInq | 1 : NoSpecify is available 0 : NoSpecify is not available This field represents whether NoSpecify mode is available or not. |

Table 8 – Inquiry (Contd.)

| Implemented | Active | Writable | Readable | Value | Description |
|-------------|----------|----------|----------|--------------------------|---|
| 0 | not-used | not-used | not-used | not-used | Device doesn't have any internal functions related to this FeatureCSR . Or device has an internal function which is not provided any control. |
| 1 | 0 | X | X | X | Device has an internal function related to this FeatureCSR , but it is not active temporarily because of other FeatureCSRs setting. (e.g. Hue is not active when PixelCoding = Mono) Host may turn it active with writing other FeatureCSRs . |
| 1 | 1 | 0 | X | last update / last write | Value is locked temporarily because of other FeatureCSRs setting. (e.g. ImageSize is locked when AcquisitionCommand = Continuous) Host may unlock Value by appropriately change the FeatureCSRs . Or Value is write-disable (read-only) permanently. |
| 1 | 1 | X | 0 | not used | Value is invalid number because host wrote to chaining ExpandedCSR , (e.g. In case that device which has chaining ExpandedCSR whose dynamic range is wider than BasicCSR , Value of BasicCSR is invalid when host wrote external value to ExpandedCSR) or host wrote to FeatureCSR which handles the same internal function in the device. (e.g. In case that device which has Gamma and LUT FeatureCSRs using unity internal function, Value of Gamma is invalid when host wrote to LUT FeatureCSR at the last). Or device is not able to indicate actual number when Control is NoSpecify , Auto or AutoOnce . |

Table 9 – Truth table of Inquiry

3.5.2 OffsetForExpanded

This field is optional. When this field is **0x00000000**, then this feature has no **ExpandedCSRs**. Chaining **ExpandedCSR** starts at top address of **CategoryBlock** + **OffsetForExpanded** value.

[CR-8cd]

OffsetForExpanded shall be multiple of quadlet (4 Bytes).

| Bit | Field | Description |
|---------|--------|--|
| [23..0] | Offset | Indicates the offset address of chaining ExpandedCSR from the top of CategoryBlock . |

Table 10 – OffsetForExpanded

3.5.3 Control

The **Control** field sets the mode which determines how to the **Value** field is handled in the **ValueRegisterArea**.

[R-9cd]

If the **Control** field is set to an unavailable control number which is specified in the **ControlInq**, device shall discard this writing number and keep previous number.

| Bit | Field | Description |
|--------|---------|---|
| [3..0] | Control | Mode control 0 : NoSpecify 1 : Manual 2 : Auto 3 : AutoOnce 4 : Default Set the Mode number among available modes. |

Table 11 – Control

3.5.3.1 NoSpecify

Device may set **Value** to device-dependent value.

[O-10cd]

Device should update **Value** to actual value.

[CR-11cd]

Device shall turn **Readable** to **0** if device cannot report actual value.

[R-12cd]

Device shall discard new values on every writing transaction when **Control** is **NoSpecify**.

3.5.3.2 Manual

Host may write **Value**.

[O-13cd]

Device should not change **Value** except in the following case:

- Corrects rounding issue
- In case of using state control, internal state was changed.
- Follows actual value updated by chaining **FeatureCSR** (please see section 5.1.1 Value).
- Follows actual value updated by **FeatureCSR** which uses same internal function.
- The range of **Value** (**Min**, **Max** and **Inc**) was changed and **Value** is out of range.

[O-14ch]

Because of previous cases, host should read back **Value** after writing.

3.5.3.3 Auto

Device adjusts **Value** automatically by itself.

[O-15cd]

Device should update **Value** whenever actual value was changed.

[CR-16cd]

Device shall turn **Readable** to **0** if device cannot report actual value.

Host may change **Control** while adjustment is running.

[O-17cd]

Device shall set **Value** to the last adjusted value if host changes **Control** from **Auto** to **Manual**,

[CR-18cd]

Device shall turn **Readable** to **0** if device cannot keep last adjusted value.

[R-19cd]

Device shall discard new values on every writing transaction when **Control** is **Auto**.

3.5.3.4 AutoOnce

Device adjusts **Value** by itself only once.

[R-20cd]

After adjustment, Device shall turn **Control** to **Manual**.

[O-21cd]

After adjustment, Device should update **Value** to actual value.

[CR-22cd]

If device cannot report adjusted **Value**, device shall turn **Readable** to **0**.

Host may change **Control** to **Manual** while adjustment is running.

[R-23cd]

If host changes **Control** to **Manual**, device shall keep previous **Value** (before set to **AutoOnce**).

[R-24cd]

Device shall discard new values on every writing transaction when **Control** is **AutoOnce**.

3.5.3.5 Default

Device changes **Value** and **Control** to default values which are device dependent.

If device default is **Auto**, the device sets **Control** to **Auto** after it receives **Default** control setting.

3.5.4 ValueRegisterArea (Common register type)

All **FeatureCSRs** have **ValueType** described in this section (common register type). **ValueType** in **BasicCSRs** is fixed in the **IIDC2** specification.

[R-25cd]

Device shall not change **ValueType** in **BasicCSRs**.

3.5.4.1 Integer

| Offset | Definitions |
|--------|-------------------------|
| +0x00 | Mult[31..0] (read-only) |
| +0x04 | Div[31..0] (read-only) |
| +0x08 | Min[31..0] (read-only) |
| +0x0C | Max[31..0] (read-only) |
| +0x10 | Value[31..0] |

Table 12 – Integer32 (ValueType = 0x30)

Big Endian

| Offset | Definitions |
|--------|--------------------------|
| +0x00 | Mult[63..32] (read-only) |
| +0x04 | Mult[31..0] (read-only) |
| +0x08 | Div[63..32] (read-only) |
| +0x0C | Div[31..0] (read-only) |
| +0x10 | Min[63..32] (read-only) |
| +0x14 | Min[31..0] (read-only) |
| +0x18 | Max[63..32] (read-only) |
| +0x1C | Max[31..0] (read-only) |
| +0x20 | Value[63..32] |
| +0x24 | Value[31..0] |

Little Endian

| Offset | Definitions |
|--------|--------------------------|
| +0x00 | Mult[31..0] (read-only) |
| +0x04 | Mult[63..32] (read-only) |
| +0x08 | Div[31..0] (read-only) |
| +0x0C | Div[63..32] (read-only) |
| +0x10 | Min[31..0] (read-only) |
| +0x14 | Min[63..32] (read-only) |
| +0x18 | Max[31..0] (read-only) |
| +0x1C | Max[63..32] (read-only) |
| +0x20 | Value[31..0] |
| +0x24 | Value[63..32] |

Table 13 – Integer64 (ValueType = 0x40)

Value is signed integer.

[R-26c]

Value shall support integers ranging from **Min** to **Max**.

Host may calculate normalized value; its unit and scale are defined by specification. This value is called **AbsoluteValue**.

[CR-27c]

If both **Mult** and **Div** are valid values, then **Value** represents **AbsoluteValue** with unit (unit is specified in 4 Device Control Register) and both **Mult** and **Div** are also signed integer.

[CR-28cd]

If **Mult** is set to **0**, **div** shall be kept to **0**. If **Div** is set to **0**, **Mult** shall be kept to **0**.

Then **AbsoluteValue** is calculated by the equation below.

$$\text{AbsoluteValue} = \text{Value} * (\text{Mult} / \text{Div}) \quad (\text{Mult} \neq \text{Div} \text{ or } \text{Mult} = \text{div} = 1)$$

Device may indicate a resolution of **Value** (not **AbsoluteValue**) by increment value (**Inc**). If **Mult = Div** and they are not equal to **0** or **1**, **Inc** is same as **Mult**. In other case, **Inc** is implicated to **1**.

$$\text{Value} = n * \text{Inc} \quad (n \text{ is integer})$$

$$\text{Inc} = \begin{cases} \text{Mult} = \text{Div} & (\text{Mult} = \text{Div} \neq 0 \text{ and } \text{Mult} = \text{Div} \neq 1) \\ 1 & (\text{Mult} = \text{Div} = 0 \text{ or } \text{Mult} = \text{Div} = 1) \end{cases}$$

[CR-29cd]

If host writes the value that is out of range from **Min** to **Max** or not multiples of **Inc**, device shall discard this writing value and keep previous value.

3.5.4.2 **PlainInteger**

Big Endian

| Offset | Definitions | |
|--------|-------------|----------|
| +0x00 | Value[7..0] | reserved |

Little Endian

| Offset | Definitions | |
|--------|-------------|-------------|
| +0x00 | reserved | Value[7..0] |

Table 14 – PlainInteger8 (ValueType = 0x11)

| Offset | Definitions |
|--------|--------------|
| +0x00 | Value[31..0] |

Table 15 – PlainInteger32 (ValueType = 0x31)

Big Endian

| Offset | Definitions |
|--------|---------------|
| +0x00 | Value[63..32] |
| +0x04 | Value[31..0] |

Little Endian

| Offset | Definitions |
|--------|---------------|
| +0x00 | Value[31..0] |
| +0x04 | Value[63..32] |

Table 16 – PlainInteger64 (ValueType = 0x41)

Value is unsigned integer, and it has no information (**Min**, **Max**, **Mult** or **Div**). **Value** may take full range of the bit width.

3.5.4.3 Float

| Offset | Definitions |
|--------|------------------------|
| +0x00 | Min[31..0] (read-only) |
| +0x04 | Max[31..0] (read-only) |
| +0x08 | Value[31..0] |

Table 17 – Float32 (ValueType = 0x32)

Big Endian

| Offset | Definitions |
|--------|-------------------------|
| +0x00 | Min[63..32] (read-only) |
| +0x04 | Min[31..0] (read-only) |
| +0x08 | Max[63..32] (read-only) |
| +0x0C | Max[31..0] (read-only) |
| +0x10 | Value[63..32] |
| +0x14 | Value[31..0] |

Little Endian

| Offset | Definitions |
|--------|-------------------------|
| +0x00 | Min[31..0] (read-only) |
| +0x04 | Min[63..32] (read-only) |
| +0x08 | Max[31..0] (read-only) |
| +0x0C | Max[63..32] (read-only) |
| +0x10 | Value[31..0] |
| +0x14 | Value[63..32] |

Table 18 – Float64 (ValueType = 0x42)

Value is floating point value which is defined by IEEE 754-2008.

[R-30cd]

If host writes a value which is outside the range of **Min** to **Max**, device shall discard this writing value and keep previous value.

[R-31cd]

If there is difference between writing value and applying value attributed to rounding error, device shall update **Value** to the actual value (or the nearest value).

3.5.4.4 Enumeration

| Big Endian | |
|------------|-------------------------------------|
| Offset | Definitions |
| +0x00 | ListOfElements[127..96] (read-only) |
| +0x04 | ListOfElements[95..64] (read-only) |
| +0x08 | ListOfElements[63..32] (read-only) |
| +0x0C | ListOfElements[31..0] (read-only) |
| +0x10 | Value[31..0] |

| Little Endian | |
|---------------|-------------------------------------|
| Offset | Definitions |
| +0x00 | ListOfElements[31..0] (read-only) |
| +0x04 | ListOfElements[63..32] (read-only) |
| +0x08 | ListOfElements[95..64] (read-only) |
| +0x0C | ListOfElements[127..96] (read-only) |
| +0x10 | Value[31..0] |

Table 19 – Enumeration (ValueType = 0x03)

This type is used to select one option from the list. Then number is set to **Value** register. (**Value** may take from **0** to **127**.) Each bit in **ListOfElements** represents Host-writable number of its value (**1** : writable, **0** : notwritable)

[R-32cd]

If host writes a value that is not writable in **ListOfElements** device shall discard this writing value and keep previous value.

The mechanism of **ListOfElements** limits for Host accessing only. Device may update a value regardless **ListOfElements**.

3.5.4.5 BulkBoolean

| Offset | Definitions |
|--------|--------------------------------|
| +0x00 | BitWritable[31..0] (read-only) |
| +0x04 | Value[31..0] |

Table 20 – BulkBoolean32 (ValueType = 0x34)

Big Endian

| Offset | Definitions |
|--------|---------------------------------|
| +0x00 | BitWritable[63..32] (read-only) |
| +0x04 | BitWritable[31..0] (read-only) |
| +0x08 | Value[63..32] |
| +0x0C | Value[31..0] |

Little Endian

| Offset | Definitions |
|--------|---------------------------------|
| +0x00 | BitWritable[31..0] (read-only) |
| +0x04 | BitWritable[63..32] (read-only) |
| +0x08 | Value[31..0] |
| +0x0C | Value[63..32] |

Table 21 – BulkBoolean64 (ValueType = 0x44)

This type is used to select one or more options from the list. One register has up to 32 (**BulkBoolean32**) / 64 (**BulkBoolean64**) options. Each bit has writable flag separately in **BitWritable** field. **0** is intended read-only (not writable), **1** is writable.

[R-33cd]

Device shall apply writing action to writable bit fields in **Value**.

3.5.4.6 Rectangle

| Offset | Definitions |
|--------|-------------------------------|
| +0x00 | MinOffsetX[31..0] (read-only) |
| +0x04 | IncOffsetX[31..0] (read-only) |
| +0x08 | MinWidth[31..0] (read-only) |
| +0x0C | IncWidth[31..0] (read-only) |
| +0x10 | TotalSizeX[31..0] (read-only) |
| +0x14 | MinOffsetY[31..0] (read-only) |
| +0x18 | IncOffsetY[31..0] (read-only) |
| +0x1C | MinHeight[31..0] (read-only) |
| +0x20 | IncHeight[31..0] (read-only) |
| +0x24 | TotalSizeY[31..0] (read-only) |
| +0x28 | OffsetX[31..0] |
| +0x2C | Width[31..0] |
| +0x30 | OffsetY[31..0] |
| +0x34 | Height[31..0] |

Table 22 – Rectangle32 (ValueType = 0x35)

This type is used for defining rectangle area.

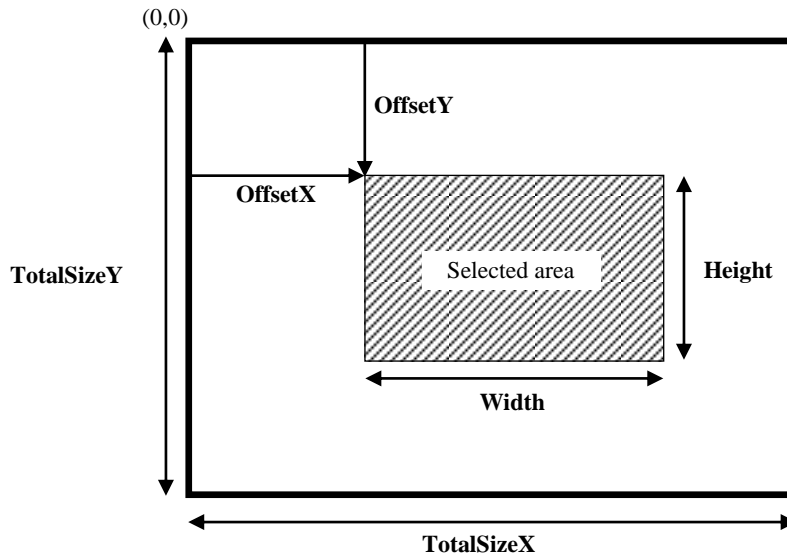


Figure 7 – Elements of Rectangle type

All values (**OffsetX**, **Width**, **OffsetY** and **Height**) are signed integer 32.

[R-34c]

Their relation shall be as follows.

$$\mathbf{OffsetX} = \mathbf{IncOffsetX} * n1 + \mathbf{MinOffsetX}$$

$$\mathbf{Width} = \mathbf{IncWidth} * n2 + \mathbf{MinWidth}$$

$$\mathbf{OffsetX} + \mathbf{Width} \leq \mathbf{TotalSizeX}$$

$$\mathbf{OffsetY} = \mathbf{IncOffsetY} * m1 + \mathbf{MinOffsetY}$$

$$\mathbf{Height} = \mathbf{IncHeight} * m2 + \mathbf{MinHeight}$$

$$\mathbf{OffsetY} + \mathbf{Height} \leq \mathbf{TotalSizeY} \quad (n1, n2, m1, m2 \text{ are integer})$$

3.5.4.7 Array of Register

All register types may be expanded to array version of that register type. **0x80** is added to the **ValueType** of standard (not arrayed) register type. For example, **ValueType** of **Integer32** is **0x30**, then **ValueType** of **ArrayOfInteger32** is **0x30 + 0x80 = 0xB0**.

In array of register, **ArrayInformationArea** is inserted on the top of **ValueRegisterArea**. And **ExtendedValueArea** is inserted under the bottom of **ValueRegisterArea**.

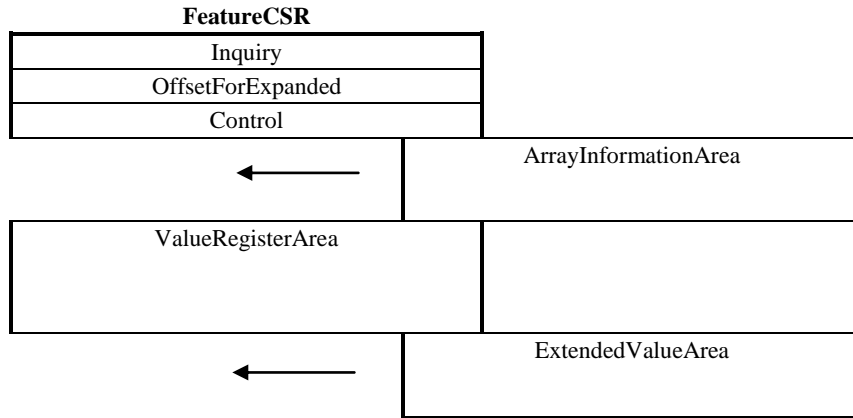


Figure 8 – Array of register

ArrayInformationArea contains one or more quadlets as described below.

| Offset | 31-24 | 23-16 | 15-8 | 7-0 |
|--------|------------------|-------|------|-----|
| +0x00 | NumberOfElements | | | |
| | MoreDimension | | | |

Table 23 – Quadlet of ArrayInformationArea

MoreDimension indicates whether next quadlet is **ArrayInformationArea** or not. If **MoreDimension = 1**, **Value** has more dimension and next quadlet indicates information of next dimension. If **MoreDimension = 0**, this quadlet is end of **ArrayInformationArea**.

NumberOfElements defines the number of elements in this dimension..

| Offset | Definitions | |
|--------|-----------------|-------------------------|
| +0x00 | MoreDimension=1 | NumberOfElements = 3 |
| +0x04 | | NumberOfElements = 2 |
| +0x08 | MoreDimension=0 | Mult[31..0] (read-only) |
| +0x0C | | Div[31..0] (read-only) |
| +0x10 | | Min[31..0] (read-only) |
| +0x14 | | Max[31..0] (read-only) |
| +0x18 | | Value[0][0][31..0] |
| +0x1C | | Value[0][1][31..0] |
| +0x20 | | Value[1][0][31..0] |
| +0x24 | | Value[1][1][31..0] |
| +0x28 | | Value[2][0][31..0] |
| +0x2C | | Value[2][1][31..0] |

Table 24 – Example of ArrayOfInteger32 with 2 dimension (ValueType = 0xB0)

3.5.4.8 String

String is substituted by **ArrayOfPlainInteger8**.

| Big Endian | | | | | |
|------------|-----------------|-----------------------|----------|----------|----------|
| Offset | Definitions | | | | |
| +0x00 | MoreDimension=0 | NumberOfElements = 16 | | | |
| +0x04 | | char[0] | char[1] | char[2] | char[3] |
| +0x08 | | char[4] | char[5] | char[6] | char[7] |
| +0x0C | | char[8] | char[9] | char[10] | char[11] |
| +0x10 | | char[12] | char[13] | char[14] | char[15] |

| Little Endian | | | | | |
|---------------|-----------------|-----------------------|----------|----------|----------|
| Offset | Definitions | | | | |
| +0x00 | MoreDimension=0 | NumberOfElements = 16 | | | |
| +0x04 | | char[3] | char[2] | char[1] | char[0] |
| +0x08 | | char[7] | char[6] | char[5] | char[4] |
| +0x0C | | char[11] | char[10] | char[9] | char[8] |
| +0x10 | | char[15] | char[14] | char[13] | char[12] |

Table 25 – Example of String with 16 characters (ValueType = 0x91)

3.5.5 Multi-Byte accessing

The data size of writing action is dependent on the interface. Then there is the situation that data size of **Value** is larger than writing data size. So host needs to use several writing actions for updating **Value**.

[R-35cd]

If host write values with several packets, device shall wait to apply the writing value until host writes to “field of Largest address number” in **Value**.

3.6 XmlManifestTable

XmlManifestTable is used to connect GenICam from IIDC2. It is used for only interfaces which have no XML Manifest in transport layer (e.g. IEEE1394).

[CR-36c]

If the interface has it in transport layer (e.g. CoaXPress), this field shall be not-used.

| Offset | Name | Description | |
|---------|---------------------|-------------|----------------------|
| +0x000 | XmlManifestSize | | |
| +0x004 | XmlManifestSelector | | |
| +0x008 | XmlVersion | [31..24] | reserved |
| | | [23..16] | XmlMajorVersion |
| | | [15..8] | XmlMinorVersion |
| | | [7..0] | XmlSubMinorVersion |
| +0x00C | XmlSchemaVersion | [31..24] | reserved |
| | | [23..16] | SchemaMajorVersion |
| | | [15..8] | SchemaMinorVersion |
| | | [7..0] | ScemaSubMinorVersion |
| +0x010~ | XmlUrlAddress | | |

Table 26 – XmlManifestTable

3.6.1 XmlManifestSize

Provides the number of XML manifests available.

3.6.2 XmlManifestSelector

Selects the required XML manifest.

[R-37c]

XMLManifestSelector shall hold a number between 0 and **XmlManifestSize**-1.

3.6.3 XmlVersion

Provides the version number for the XML file given in the manifest referenced by **XmlManifestSelector**.

3.6.4 XmlSchemaVersion

Provides the GenICam schema version for the XML file given in the manifest referenced by **XmlManifestSelector**.

3.6.5 XmlUrlAddress

Provides the address of the start of the URL string referenced by **XmlManifestSelector**.

[R-38c]

The string of **XmlUrlAddress** shall be in the format defined as follows.

3.6.5.1 URL Format - Non-Volatile Memory

[CR-39c]

If the XML files is stored in non-volatile memory in the device, the URL shall be of the form:

“Local:<Filename>.<Extension>;<Address>;<Length>” as defined in Table 27.

| Field | Description |
|-------------|---|
| Local | Indicates the XML file is stored in non-volatile memory in the device. |
| <Filename> | the name of the XML file. |
| <Extension> | “xml” indicates a text XML file (i.e. uncompressed). “zip” indicates a ZIP format compressed file. |
| <Address> | The absolute address of the XML file. It is given in hexadecimal form without a leading “0x”. |
| <Length> | The length of the XML file in bytes, given in hexadecimal without a leading “0x”. |

Table 27 – URL format – Non-volatile memory

[CO-40c]

Filename should include the vendor name, model name and device revision.

Example: “Local:MyCompany_MyProduct_Rev1.zip;B8000;33A” is a ZIP file starting at address 0xB8000 in the device with a length of 0x33A bytes. The XML file is for revision 1 of a device called “MyProduct” made by “MyCompany”.

3.6.5.2 URL Format – Vendor website

If the XML file is stored on the vendor’s website, the URL SHALL be of the form:

“Web:<WebURL>/<Filename>.<Extension>” as defined in Table 28.

| Field | Description |
|-------------|---|
| Web | Indicates the XML file is stored on the vendor’s website |
| <WebURL> | A full web URL, form the scheme name (e.g. http) to the path. |
| <Filename> | The name of the XML file. |
| <Extension> | “xml” indicates a text XML file (i.e. uncompressed). “zip” indicates a ZIP format compressed file. |

Table 28 – URL format – Vendor website

[CO-41c]

Filename should include the vendor name, model name and device revision.

Example: “Web:http://www.mycompany.com/xml/MyCompany_MyProduct_Rev1.xml” is a text XML file found at http://www.mycompany.com/xml. The XML file is for revision 1 of a device called “MyProduct” made by “MyCompany”

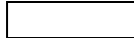


None of the fields are case sensitive.

4 Device Control Register

All tables of **CategoryBlocks** are defined as follows.

[R-42c]

In the tables of **CategoryBlocks**, Host / Device shall follow the rule as defined below.

-  Device dependent : Device may use this field.
-  Feature dependent : Device and host shall keep the constant value described in this specification.
-  reserved / not-used : Device shall keep zeros. Host shall not set ones.

If **FeatureCSR** has a **Control** and/or **Integer32** register with **AbsoluteValue**, device should comply with following table.

| | | |
|------------------------------|-----------------|---|
| Unit of AbsoluteValue | | describes unit of the AbsoluteValue |
| Reference point | | describes reference point of the AbsoluteValue |
| Recommended Value at | AutoOnce / Auto | describe recommended behavior of Value in each Controls |
| | NoSpecify | |

If **Implemented** is **0**, device may set all fields including Feature dependent field to zeros in its **FeatureCSR**.

[CR-43ch]

Host shall not refer all field except **Implement** in **FeatureCSR** if **Implement** is 0.

4.1 CategoryBlock0 (DeviceControl)

| Offset | Name | Field | Bit | Description |
|------------------|-----------------|---------------------|----------|----------------------|
| +0x000 | Header | CategoryBlockNumber | [31..24] | = 0 |
| | | SizeOfCategoryBlock | [23..0] | = 0x000160 |
| +0x004~ 0x01C | | - | - | reserved |
| +0x020 | Device Reset | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | |
| | | Readable | [24] | |
| | | ValueType | [23..16] | = 0x03 (Enumeration) |
| | | ControlInq | [15..0] | not-used |
| +0x024 | | OffsetForExpanded | [31..0] | |
| +0x028 | | Control | [31..0] | = 0x01 (Manual) |
| +0x02C~ 0x038 | ListofElements | | [127] | VendorSpecific7 |
| | | ... | ... | ... |
| | | | [121] | VendorSpecific1 |
| | | | [120] | VendorSpecific0 |
| | | | [119..2] | not-used |
| | | | [1] | Reset |
| | | | [0] | Off (Always 0) |
| +0x03C | | Value | [31..0] | |
| +0x040 | Device Power | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | |
| | | Readable | [24] | |
| | | ValueType | [23..16] | = 0x03 (Enumeration) |
| | | ControlInq | [15..0] | not-used |
| +0x044 | | OffsetForExpanded | [31..0] | |
| +0x048 | | Control | [31..0] | = 0x01 (Manual) |
| +0x04C~ 0x058 | ListofElements | | [127] | VendorSpecific7 |
| | | ... | ... | ... |
| | | | [121] | VendorSpecific1 |
| | | | [120] | VendorSpecific0 |
| | | | [119..2] | not-used |
| | | | [1] | On |
| | | | [0] | LowPower |
| +0x05C | | Value | [31..0] | |

Table 29 –CategoryBlock0 (DeviceControl)

| Offset | Name | Field | Bit | Description |
|--------------|--------------------------|------------------------------------|----------|-------------------------------|
| +0x060 | Device Vendor Name | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | = 0 (Read-only) |
| | | Readable | [24] | |
| | | ValueType | [23..16] | = 0x91 (ArrayOfPlainInteger8) |
| | | ControlInq | [15..0] | not-used |
| +0x064 | | OffsetForExpanded | [31..0] | |
| +0x068 | | Control | [31..0] | = 0x01 (Manual) |
| +0x06C | | MoreDimension | [31] | = 0 |
| | | NumberOfElements | [30..0] | = 16 |
| +0x070~0x07C | | Value[0]~Value[15] | | |
| +0x080~0x09C | Device Model Name | Same Structure as DeviceVendorName | | |
| +0x0A0~0x0BC | Device Manufacturer Info | Same Structure as DeviceVendorName | | |
| +0x0C0~0x0DC | Device Version | Same Structure as DeviceVendorName | | |
| +0x0E0~0x0FC | Device Firmware Version | Same Structure as DeviceVendorName | | |
| +0x100~0x11C | DeviceID | Same Structure as DeviceVendorName | | |
| +0x120 | Device UserID | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | |
| | | Readable | [24] | |
| | | ValueType | [23..16] | = 0x91 (ArrayOfPlainInteger8) |
| | | ControlInq | [15..0] | not-used |
| +0x124 | | OffsetForExpanded | [31..0] | |
| +0x128 | | Control | [31..0] | = 0x01 (Manual) |
| +0x12C | | MoreDimension | [31] | = 0 |
| | | NumberOfElements | [30..0] | = 16 |
| +0x130~0x13C | | Value[0]~Value[15] | | |

Table 29 –CategoryBlock0 (DeviceControl) (Contd.)

| Offset | Name | Field | Bit | Description | | |
|--------|-------------------------|-----------------|----------|--------------------|---------|-----------------|
| +0x140 | LED Indicator Luminance | Implemented | [31] | | | |
| | | Active | [30] | | | |
| | | - | [29..27] | reserved | | |
| | | UserSetLoadable | [26] | | | |
| | | Writable | [25] | | | |
| | | Readable | [24] | | | |
| | | ValueType | [23..16] | = 0x30 (Integer32) | | |
| | | ControlInq | [15..0] | not-used | | |
| | | +0x144 | | OffsetForExpanded | [31..0] | |
| | | +0x148 | | Control | [31..0] | = 0x01 (Manual) |
| +0x14C | | Mult | [31..0] | | | |
| +0x150 | | Div | [31..0] | | | |
| +0x154 | | Min | [31..0] | | | |
| +0x158 | | Max | [31..0] | | | |
| +0x15C | | Value | [31..0] | | | |

Table 29 –CategoryBlock0 (DeviceControl) (Contd.)

DeviceControl handles condition of device and indicates information of device. If the transport layer has same registers (e.g. CoaXPress has **DeviceVendorName**, **DeviceModelName**, **DeviceManufacturerInfo**, **DeviceVersion**, **DeviceFirmwareVersion**, **DeviceID** and **DeviceUserID** registers), device may not have these **FeatureCSRs**.

The information **FeatureCSRs** (from **DeviceVendorName** to **DeviceUserID**) have **Values** as ASCII strings, they are up to 16 characters.

[CR-44cd]

If length of ASCII string is less than 16 characters, device shall fill Null (0x00) character for remaining area.

Device may have **Value** beyond 16 characters by using the chaining **ExpandedCSR**.

[CR-45c]

If the information **FeatureCSRs** use **ExpandedCSRs**, **Value** of **BasicCSR** shall be filled by NULL string.

4.1.1 DeviceReset

[R-46cd]

When this feature is executed, the device behavior shall be the same transition to the power on state.

Reset: Reset the device.

Off: Normal operation.

VendorSpecific0~7 :Camera vendor specific pattern

[R-47cd]

After reset sequence, device shall turn to **Off** automatically.

[R-48ch]

Host shall not set **Off** this **Control**.

4.1.2 DevicePower

Controls device's Power.

On: Power on the device.

LowPower : Power off the device without interface block.

VendorSpecific0~7 :Camera vendor specific pattern

4.1.3 DeviceVendorName

Indicates the manufacturer's name of the device.

4.1.4 DeviceModelName

Indicates the model name of the device.

4.1.5 DeviceManufacturerInfo

Indicates the extended information of manufacturer.

4.1.6 DeviceVersion

Indicates the version of the device.

4.1.7 DeviceFirmwareVersion

Indicates the version of the firmware in the device.

4.1.8 DeviceID

Indicates the device identifier.

[O-49cd]

DeviceID should be the same as the serial number.

4.1.9 DeviceUserID

Handles the user-programmable identifier. The device user may set and read the original ID to device.

[CR-50cd]

Device shall store it to non-volatile memory in the device when this **FeatureCSR** was written.

4.1.10 LEDIndicatorLuminance

Handles the luminance of LED indicator. It is used when the device has LED for indicate device status (Power supply, data transfer or etc...).

| | | |
|-------------------------------|-----------|-----------------------|
| Unit of Absolute Value | | % |
| Reference point | | 0 |
| Recommended value at | NoSpecify | Factory setting value |

4.2 CategoryBlock1 (TransportLayerControl)

This **CategoryBlock** is different for each interfaces.

Please see IIDC2 Ver.1.0.0 (for IEEE1394) or each specification of transport layer.

4.3 CategoryBlock2 (ImageFormatControl)

| Offset | Name | Field | Bit | Description |
|------------------|-----------------------------|---------------------|-----------|----------------------|
| +0x000 | Header | CategoryBlockNumber | [31..24] | = 2 |
| | | SizeOfCategoryBlock | [23..0] | = 0x000200 |
| +0x004~ 0x01C | | - | - | reserved |
| +0x020 | Image Format Selector | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | |
| | | Readable | [24] | |
| | | ValueType | [23..16] | = 0x03 (Enumeration) |
| | | - | [15..5] | reserved |
| | | DefaultInq | [4] | |
| | | AutoOnceInq | [3] | not-used |
| | | AutoInq | [2] | not-used |
| | | ManualInq | [1] | |
| | | NoSpecifyInq | [0] | not-used |
| | | +0x024 | | OffsetForExpanded |
| +0x028 | | - | [31..4] | reserved |
| | | Control | [3..0] | |
| +0x02C~ 0x038 | ListOfElements | | [127] | VendorSpecific7 |
| | | ... | ... | ... |
| | | | [121] | VendorSpecific1 |
| | | | [120] | VendorSpecific0 |
| | | | [119..32] | not-used |
| | | | [31] | Format31 |
| | | ... | ... | ... |
| | | | [1] | Format1 |
| | [0] | Format0 | | |
| +0x03C | | Value | [31..0] | |
| +0x040 | Apply Image Format | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | |
| | | Readable | [24] | |
| | | ValueType | [23..16] | = 0x03 (Enumeration) |
| | | ControlInq | [15..0] | not-used |
| +0x044 | | OffsetForExpanded | [31..0] | |
| +0x048 | | Control | [31..0] | = 0x01 (Manual) |

Table 30 – CategoryBlock2 (ImageFormatControl)

| Offset | Name | Field | Bit | Description |
|------------------|--------------------------|-----------------|-----------|-----------------------------|
| +0x04C~ 0x058 | Apply Image Format | ListOfElements | [127..17] | not-used |
| | | | [16] | ImageFormatError (Always 0) |
| | | | [15..9] | not-used |
| | | | [8] | Changed (Always 0) |
| | | | [7..2] | not-used |
| | | | [1] | Apply |
| +0x05C | | Value | [31..0] | |
| +0x060 | ImageSize | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | |
| | | Readable | [24] | |
| | | ValueType | [23..16] | = 0x35 (Rectangle32) |
| | | - | [15..5] | reserved |
| | | DefaultInq | [4] | |
| | | AutoOnceInq | [3] | not-used |
| | | AutoInq | [2] | not-used |
| | | ManualInq | [1] | |
| | | NoSpecifyInq | [0] | not-used |
| | | +0x064 | | OffsetForExpanded |
| +0x068 | | - | [31..4] | reserved |
| | | Control | [3..0] | |
| +0x06C | | MinOffsetX | [31..0] | |
| +0x070 | | IncOffsetX | [31..0] | |
| +0x074 | | MinWidth | [31..0] | |
| +0x078 | | IncWidth | [31..0] | |
| +0x07C | | TotalSizeX | [31..0] | |
| +0x080 | | MinOffsetY | [31..0] | |
| +0x084 | | IncOffsetY | [31..0] | |
| +0x088 | | MinHeight | [31..0] | |
| +0x08C | | IncHeight | [31..0] | |
| +0x090 | | TotalSizeY | [31..0] | |
| +0x094 | | OffsetX | [31..0] | |
| +0x098 | | Width | [31..0] | |
| +0x09C | | OffsetY | [31..0] | |
| +0x0A0 | | Height | [31..0] | |
| +0x0A4~ 0x0BC | | - | - | reserved |

Table 30 – CategoryBlock2 (ImageFormatControl) (Contd.)

| Offset | Name | Field | Bit | Description |
|------------------|----------------|-------------------|------------|----------------------|
| +0x0C0 | PixelCoding | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | |
| | | Readable | [24] | |
| | | ValueType | [23..16] | = 0x03 (Enumeration) |
| | | - | [15..5] | reserved |
| | | DefaultInq | [4] | |
| | | AutoOnceInq | [3] | not-used |
| | | AutoInq | [2] | not-used |
| | | ManualInq | [1] | |
| | | NoSpecifyInq | [0] | not-used |
| +0x0C4 | | OffsetForExpanded | [31..0] | |
| +0x0C8 | | - | [31..4] | reserved |
| | | Control | [3..0] | |
| +0x0CC~ 0x0D8 | ListOfElements | | [127..108] | not-used |
| | | | [107] | BayerBGPacked |
| | | | [106] | not-used |
| | | | [105] | BayerBG |
| | | | [104] | BayerGBPacked |
| | | | [103] | not-used |
| | | | [102] | BayerGB |
| | | | [101] | BayerRGPacked |
| | | | [100] | not-used |
| | | | [99] | BayerRG |
| | | | [98] | BayerGRPacked |
| | | | [97] | not-used |
| | | | [96] | BayerGR |
| | | | [95..83] | not-used |
| | | | [82] | YUV444Packed |
| | | | [81..75] | not-used |
| | | | [74] | YUV422Packed |
| | | | [73..67] | not-used |
| | | | [66] | YUV411Packed |
| | | | [65..43] | not-used |
| | | | 42 | BGRPacked |
| | | | [41..35] | not-used |
| | | | [34] | RGBPacked |
| | | | [33..3] | not-used |
| | | | [2] | MonoPacked |
| | | | [1] | MonoSigned |
| | | | [0] | Mono |
| +0x0DC | | Value | [31..0] | |

Table 30 – CategoryBlock2 (ImageFormatControl) (Contd.)

| Offset | Name | Field | Bit | Description |
|------------------|----------------|-------------------|-----------|----------------------|
| +0x0E0 | PixelFormat | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | |
| | | Readable | [24] | |
| | | ValueType | [23..16] | = 0x03 (Enumeration) |
| | | - | [15..5] | reserved |
| | | DefaultInq | [4] | |
| | | AutoOnceInq | [3] | not-used |
| | | AutoInq | [2] | not-used |
| | | ManualInq | [1] | |
| | | NoSpecifyInq | [0] | not-used |
| +0x0E4 | | OffsetForExpanded | [31..0] | |
| +0x0E8 | | - | [31..4] | reserved |
| | | Control | [3..0] | |
| +0x0EC~ 0x0F8 | ListofElements | | [127..49] | not-used |
| | | | [48] | Bpp48 |
| | | | [47..25] | not-used |
| | | | [24] | Bpp24 |
| | | | [23..17] | not-used |
| | | | [16] | Bpp16 |
| | | | [15] | not-used |
| | | | [14] | Bpp14 |
| | | | [13] | not-used |
| | | | [12] | Bpp12 |
| | | | [11] | not-used |
| | | | [10] | Bpp10 |
| | | | [9] | not-used |
| | | | [8] | Bpp8 |
| | [7..0] | not-used | | |
| +0x0FC | | Value | [31..0] | |

Table 30 – CategoryBlock2 (ImageFormatControl) (Contd.)

| Offset | Name | Field | Bit | Description |
|------------------|-----------------------|-----------------|----------|---|
| +0x100 | PixelEndian | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | |
| | | Readable | [24] | |
| | | ValueType | [23..16] | = 0x03 (Enumeration) |
| | | - | [15..5] | reserved |
| | | DefaultInq | [4] | |
| | | AutoOnceInq | [3] | not-used |
| | | AutoInq | [2] | not-used |
| | | ManualInq | [1] | |
| | | NoSpecifyInq | [0] | not-used |
| | | +0x104 | | OffsetForExpanded |
| +0x108 | | - | [31..4] | reserved |
| | | Control | [3..0] | |
| +0x10C~ 0x118 | | ListOfElements | [127..2] | not-used |
| | | | [1] | LittleEndian |
| | | | [0] | BigEndian |
| +0x11C | | Value | [31..0] | |
| +0x120 | Binning Horizontal | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | |
| | | Readable | [24] | |
| | | ValueType | [23..16] | = 0x30 (Integer32) |
| | | - | [15..5] | reserved |
| | | DefaultInq | [4] | |
| | | AutoOnceInq | [3] | not-used |
| | | AutoInq | [2] | not-used |
| | | ManualInq | [1] | |
| | | NoSpecifyInq | [0] | not-used |
| | | +0x124 | | OffsetForExpanded |
| +0x128 | | - | [31..4] | reserved |
| | | Control | [3..0] | |
| +0x12C | | Mult | [31..0] | Inc BinningHorizontal |
| +0x130 | | Div | [31..0] | (These field SHALL be used as Inc only) |
| +0x134 | | Min | [31..0] | |
| +0x138 | | Max | [31..0] | |
| +0x13C | | Value | [31..0] | |

Table 30 – CategoryBlock2 (ImageFormatControl) (Contd.)

| Offset | Name | Field | Bit | Description | | |
|------------------|--------------------------|-------------------------------------|----------|------------------------|---------|----------|
| +0x140~ 0x15C | Binning Vertical | Same Structure as BinningHorizontal | | | | |
| +0x160~ 0x17C | Decimation Horizontal | Same Structure as BinningHorizontal | | | | |
| +0x180~ 0x19C | Decimation Vertical | Same Structure as BinningHorizontal | | | | |
| +0x1A0 | ReverseX | Implemented | [31] | | | |
| | | Active | [30] | | | |
| | | - | [29..27] | reserved | | |
| | | UserSetLoadable | [26] | | | |
| | | Writable | [25] | | | |
| | | Readable | [24] | | | |
| | | ValueType | [23..16] | = 0x34 (BulkBoolean32) | | |
| | | - | [15..5] | reserved | | |
| | | DefaultInq | [4] | | | |
| | | AutoOnceInq | [3] | not-used | | |
| | | AutoInq | [2] | not-used | | |
| | | ManualInq | [1] | | | |
| | | NoSpecifyInq | [0] | not-used | | |
| | | +0x1A4 | | OffsetForExpanded | [31..0] | |
| | | +0x1A8 | | - | [31..4] | reserved |
| | Control | | [3..0] | | | |
| +0x1AC | | BitWritable | [31..1] | not-used | | |
| | | | [0] | | | |
| +0x1B0 | | Value | [31..0] | not-used | | |
| | | | [0] | | | |
| +0x1B4~ 0x1BC | | - | - | reserved | | |
| +0x1C0~ 0x1DC | ReverseY | Same Structure as ReverseX | | | | |

Table 30 – CategoryBlock2 (ImageFormatControl) (Contd.)

| Offset | Name | Field | Bit | Description | | |
|------------------|-------------|-----------------|-----------|--------------------------|---------|--|
| +0x1E0 | TestPattern | Implemented | [31] | | | |
| | | Active | [30] | | | |
| | | - | [29..27] | reserved | | |
| | | UserSetLoadable | [26] | | | |
| | | Writable | [25] | | | |
| | | Readable | [24] | | | |
| | | ValueType | [23..16] | = 0x03 (Enumeration) | | |
| | | - | [15..5] | reserved | | |
| | | DefaultInq | [4] | | | |
| | | AutoOnceInq | [3] | not-used | | |
| | | AutoInq | [2] | not-used | | |
| | | ManualInq | [1] | | | |
| | | NoSpecifyInq | [0] | not-used | | |
| | | +0x1E4 | | OffsetForExpanded | [31..0] | |
| +0x1E8 | | - | [31..4] | reserved | | |
| +0x1EC~ 0x1F8 | | Control | [3..0] | | | |
| | | ListOfElements | [127] | VendorSpecific7 | | |
| | | ... | ... | ... | | |
| | | | [121] | VendorSpecific1 | | |
| | | | [120] | VendorSpecific0 | | |
| | | | [119..50] | not-used | | |
| | | | [49] | GreyScaleChart | | |
| | | | [48] | ColorBar | | |
| | | | [47..42] | not-used | | |
| | | | [41] | VerticalLineMoving | | |
| | | | [40] | HorizontalLineMoving | | |
| | | | [39..34] | not-used | | |
| | | | [33] | GreyVerticalRampMoving | | |
| | | | [32] | GreyHorizontalRampMoving | | |
| | | | [31..25] | not-used | | |
| | | | [24] | StraightCounter | | |
| | | | [23..18] | not-used | | |
| | | | [17] | GreyVerticalRamp | | |
| | | | [16] | GreyHorizontalRamp | | |
| | | | [15..5] | not-used | | |
| | | | [4] | GreyAA | | |
| | | | [3] | Grey55 | | |
| | | | [2] | White | | |
| | | | [1] | Black | | |
| | | | [0] | Off | | |
| | | +0x1FC | | Value | [31..0] | |

4.3.1 ImageFormatSelector

Device may have several image formats. **ImageFormatSelector** handles which format will be active.

[R-51cd]

If **ImageFormatSelector** is changed, device shall switch attached **FeatureCSRs** (**ImageSize**, **PixelCoding** **PixelSize** and **PixelEndian**) for active format.

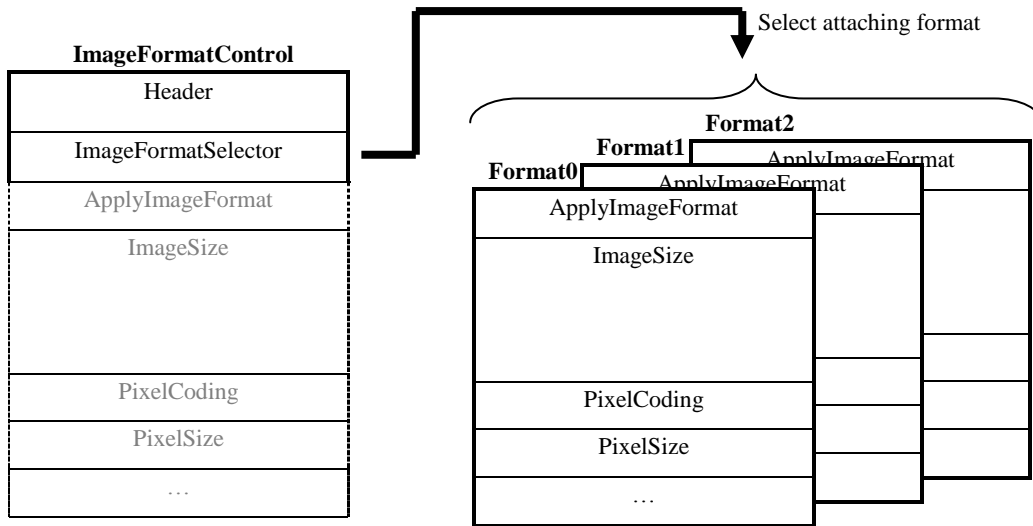


Figure 9 – ImageFormatSelector

4.3.2 ApplyImageFormat

Purpose of this **FeatureCSR** is for applying the image format setting (**ImageSize**, **PixelCoding** and **PixelSize FeatureCSRs**) to **TransportLayerControl**.

Done : Indicates image format setting is already applied.

Apply :Applies image format setting.

Changed : Indicates image format setting was changed since last **Apply** state was executed.

ImageFormatError : Indicates the image format setting has an error.

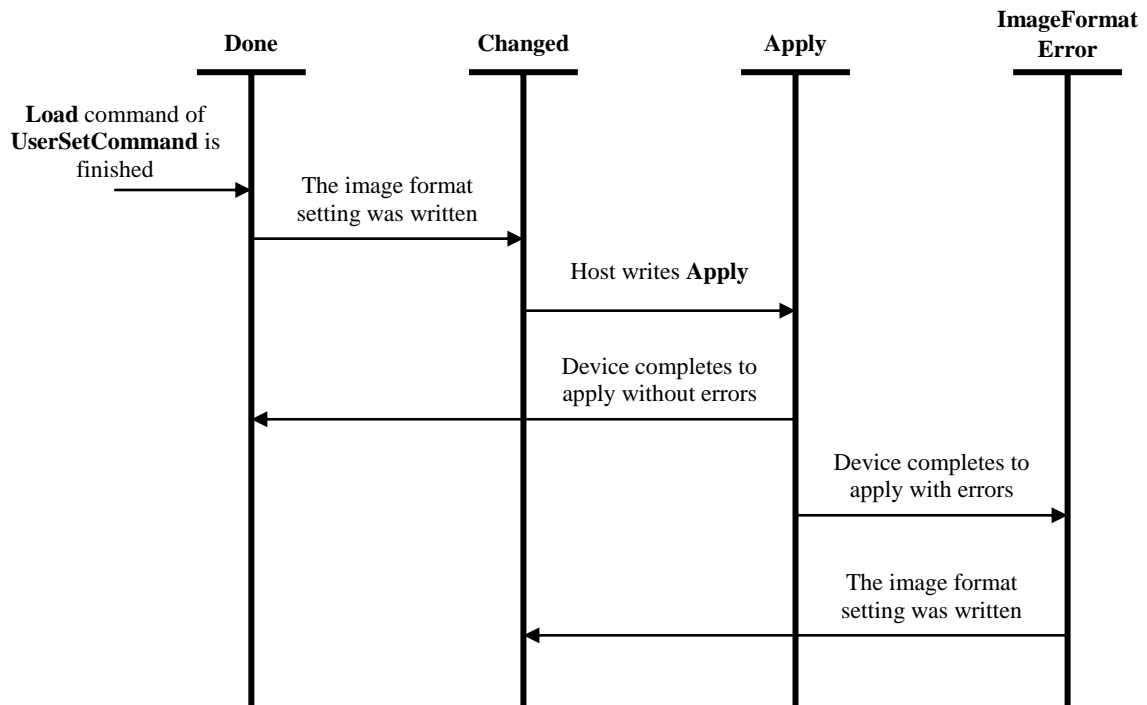


Figure 10 – State machine of ApplyImageFormat

[R-52ch]

Host shall not set **Done**, **Changed** or **ImageFormatError** for **Control**.

[CR-53cd]

If applying sequence is finished without any error, device shall change the **Control** to **Done**.

[R-54cd]

If applying sequence is finished with error, device shall change the **Control** to **ImageFormatError**.

[R-55cd]

If image format setting is changed, device shall move to **Changed**.

If device supports immediate applying, device may implement **Done** and **ImageFormatError** command only as read-only register (**Writable** is set to zero).

4.3.3 ImageSize

Specifies the Image size.

4.3.4 PixelCoding / PixelSize

These features describe available pixel format capability of the device. It is called **PixelFormat** virtually. Available combination of these and corresponding **PixelFormat** are as followed.

| PixelCoding | PixelSize | PixelFormat |
|-----------------------|--------------|-----------------|
| Mono (= 0) | Bpp8 (= 8) | Mono8 |
| Mono (= 0) | Bpp10 (= 10) | Mono10 |
| Mono (= 0) | BPP12 (= 12) | Mono12 |
| MonoPacked (= 2) | BPP12 (= 12) | Mono12Packed |
| YUV411Packed (= 66) | BPP12 (= 12) | YUV411Packed |
| YUV422Packed (= 74) | Bpp16 (= 16) | YUV422Packed |
| YUV444Packed (= 82) | Bpp24 (= 24) | YUV444Packed |
| RGBPacked (= 34) | Bpp24 (= 24) | RGB8Packed |
| BGRPacked(= 42) | Bpp24 (= 24) | BGR8Packed |
| BayerGR (= 96) | Bpp8 (= 8) | BayerGR8 |
| BayerGR (= 96) | Bpp8 (= 10) | BayerGR10 |
| BayerGR (= 96) | Bpp12 (= 12) | BayerGR12 |
| BayerGRPacked (= 98) | BPP12 (= 12) | BayerGR12Packed |
| BayerRG (= 99) | Bpp8 (= 8) | BayerRG8 |
| BayerRG (= 99) | Bpp8 (= 10) | BayerRG10 |
| BayerRG (= 99) | Bpp12 (= 12) | BayerRG12 |
| BayerRGPacked (= 101) | BPP12 (= 12) | BayerRG12Packed |
| BayerGB (= 102) | Bpp8 (= 8) | BayerGB8 |
| BayerGB (= 102) | Bpp8 (= 10) | BayerGB10 |
| BayerGB (= 102) | Bpp12 (= 12) | BayerGB12 |
| BayerGBPacked (= 104) | BPP12 (= 12) | BayerGB12Packed |
| BayerBG (= 105) | Bpp8 (= 8) | BayerBG8 |
| BayerBG (= 105) | Bpp8 (= 10) | BayerBG10 |
| BayerBG (= 105) | Bpp12 (= 12) | BayerBG12 |
| BayerBGPacked (= 107) | BPP12 (= 12) | BayerBG12Packed |

Table 31 – Pixel Coding, PixelSize and corresponding PixelFormat

PixelCoding is primary setting value, and **PixelSize** is secondary. Range of **PixelSize** is limited by **PixelCoding**.

[R-56ch]

Host shall set **PixelCoding** first when it wants to change **PixelFormat**.

4.3.5 PixelEndian

Selects Endian of pixel data.

4.3.6 BinningHorizontal / BinningVertical

Sets pixel count to combine together.

4.3.7 DecimationHorizontal / DecimationVertical

Sets sub-sampling count of the image.

4.3.8 ReverseX / ReverseY

Flip horizontal / vertical image.

4.3.9 TestPattern

Selects test pattern.

Off : Test pattern generator is turn off. Host can get the image from sensor.

Black : filled with black (0x00@8 bit luminance) pixel

White : filled with white (0xFF@8 bit luminance) pixel

Grey55 : filled with Grey (0x55@8 bit luminance) pixel

GreyAA : filled with Grey (0xAA@8 bit luminance) pixel

Black : filled with black (0x00@8 bit luminance)

GreyHorizontalRamp : Horizontal ramp image

GreyVerticalRamp : Vertical ramp image

StraightCounter : Simple counter unrelated with width / height

ColorBar : Color bar chart including Black, Blue, Red, Magenta, Green, Cyan, Yellow, White

GreyScaleChart : Grey scale chart including a few luminance grey region

VendorSpecific0~7 : Camera vendor specific pattern

4.4 CategoryBlock3 (AcquisitionControl)

| Offset | Name | Field | Bit | Description | | |
|------------------|---------------------------|---------------------|-----------|----------------------|---------|-----------------|
| +0x000 | Header | CategoryBlockNumber | [31..24] | = 3 | | |
| | | SizeOfCategoryBlock | [23..0] | = 0x0000E0 | | |
| +0x004~ 0x01C | | - | - | reserved | | |
| +0x020 | Acquisition Command | Implemented | [31] | | | |
| | | Active | [30] | | | |
| | | - | [29..27] | reserved | | |
| | | UserSetLoadable | [26] | | | |
| | | Writable | [25] | | | |
| | | Readable | [24] | | | |
| | | ValueType | [23..16] | = 0x03 (Enumeration) | | |
| | | ControlInq | [15..0] | not-used | | |
| | | +0x024 | | OffsetForExpanded | [31..0] | |
| | | +0x028 | | Control | [31..0] | = 0x01 (Manual) |
| +0x02C~ 0x038 | ListofElements | | [127..17] | not-used | | |
| | | | [16] | Retransmit | | |
| | | | [15..11] | not-used | | |
| | | | [10] | ImageBufferRead | | |
| | | | [9] | MultiFrame | | |
| | | | [8] | Continuous | | |
| | | | [7..2] | not-used | | |
| | | | [1] | Stop | | |
| | | | [0] | Abort | | |
| +0x03C | | Value | [31..0] | | | |
| +0x040 | Acquisition FrameCount | Implemented | [31] | | | |
| | | Active | [30] | | | |
| | | - | [29..27] | reserved | | |
| | | UserSetLoadable | [26] | | | |
| | | Writable | [25] | | | |
| | | Readable | [24] | | | |
| | | ValueType | [23..16] | = 0x30 (Integer32) | | |
| | | ControlInq | [15..0] | not-used | | |
| | | +0x044 | | OffsetForExpanded | [31..0] | |
| | | +0x048 | | Control | [31..0] | = 0x01 (Manual) |
| +0x04C | | Mult | [31..0] | not-used | | |
| +0x050 | | Div | [31..0] | not-used | | |
| +0x054 | | Min | [31..0] | | | |
| +0x058 | | Max | [31..0] | | | |
| +0x05C | | Value | [31..0] | | | |

Table 32 – CategoryBlock3 (AcquisitionControl)

| Offset | Name | Field | Bit | Description | |
|------------------|------------------------|-----------------|-------------------|----------------------|-----------------|
| +0x060 | ImageBuffer Mode | Implemented | [31] | | |
| | | Active | [30] | | |
| | | - | [29..27] | reserved | |
| | | UserSetLoadable | [26] | | |
| | | Writable | [25] | | |
| | | Readable | [24] | | |
| | | ValueType | [23..16] | = 0x03 (Enumeration) | |
| | | ControlInq | [15..0] | not-used | |
| | | +0x064 | OffsetForExpanded | [31..0] | |
| | | +0x068 | Control | [31..0] | = 0x01 (Manual) |
| +0x06C~ 0x078 | ListOfElements | | [127..2] | not-used | |
| | | | [1] | On | |
| | | | [0] | Off | |
| +0x07C | Value | [31..0] | | | |
| +0x080 | ImageBuffer FrameCount | Implemented | [31] | | |
| | | Active | [30] | | |
| | | - | [29..27] | reserved | |
| | | UserSetLoadable | [26] | | |
| | | Writable | [25] | | |
| | | Readable | [24] | = 0 (read-only) | |
| | | ValueType | [23..16] | = 0x30 (Integer32) | |
| | | ControlInq | [15..0] | not-used | |
| | | +0x084 | OffsetForExpanded | [31..0] | |
| | | +0x088 | Control | [31..0] | = 0x01 (Manual) |
| | | +0x08C | Mult | [31..0] | not-used |
| | | +0x090 | Div | [31..0] | not-used |
| | | +0x094 | Min | [31..0] | = 0 (Always 0) |
| | | +0x098 | Max | [31..0] | |
| +0x09C | Value | [31..0] | | | |
| +0x0A0 | Acquisition FrameRate | Implemented | [31] | | |
| | | Active | [30] | | |
| | | - | [29..27] | reserved | |
| | | UserSetLoadable | [26] | | |
| | | Writable | [25] | | |
| | | Readable | [24] | | |
| | | ValueType | [23..16] | = 0x30 (Integer32) | |
| | | ControlInq | [15..5] | not-used | |
| | | DefaultInq | [4] | | |
| | | AutoOnceInq | [3] | not-used | |
| | | AutoInq | [2] | not-used | |
| | | ManualInq | [1] | | |
| | | NoSpecifyInq | [0] | | |

Table 32 – CategoryBlock3 (AcquisitionControl) (Contd.)

| Offset | Name | Field | Bit | Description |
|------------------|----------------------------------|--|---------|-------------|
| +0x0A4 | Acquisition FrameRate | OffsetForExpanded | [31..0] | |
| +0x0A8 | | - | [31..4] | reserved |
| | | Control | [3..0] | |
| +0x0AC | | Mult | [31..0] | |
| +0x0B0 | | Div | [31..0] | |
| +0x0B4 | | Min | [31..0] | |
| +0x0B8 | | Max | [31..0] | |
| +0x0BC | | Value | [31..0] | |
| +0x0C0~ 0x0DC | Acquisition Frame Interval | Same Structure as AcquisitionFrameRate | | |

Table 32 – CategoryBlock3 (AcquisitionControl) (Contd.)

4.4.1 AcquisitionCommand

Handles to capture image and transfer image data. Image Buffer feature may be also controlled.

Stop : Stop capturing and transferring data after current transmitting frame is finished.

Abort: Abort capturing and transferring data immediately.

Continuous : Start capturing and transferring data continuously.

MultiFrame : Start capturing and transferring multiple frames. The number of capturing / transferring frames are defined by **AcquisitionFrameCount**. After transmission is finished, this field turns to **Stop** or **Abort** automatically.

ImageBufferRead: Transfer image data from Image Buffer. It is available until **ImageBufferMode** = On. The number of transferring frames are defined by **AcquisitionFrameCount**. After transmission is finished, this field turns to **Stop** or **Abort** automatically.

Retransmit : Retransmit image data transferred last. After transmission is finished, this field turns to **Stop** or **Abort** automatically.

4.4.2 AcquisitionFrameCount

Sets number of capturing / transferring frames.

4.4.3 ImageBufferMode

Handles Image Buffer. This feature is related to **AcquisitionCommand**.

Off : Disable Image Buffer feature.

On : Capture and store image data into Image Buffer.

The block diagram of **AcquisitionCommand** and **ImageBufferMode** is shown as following figure.

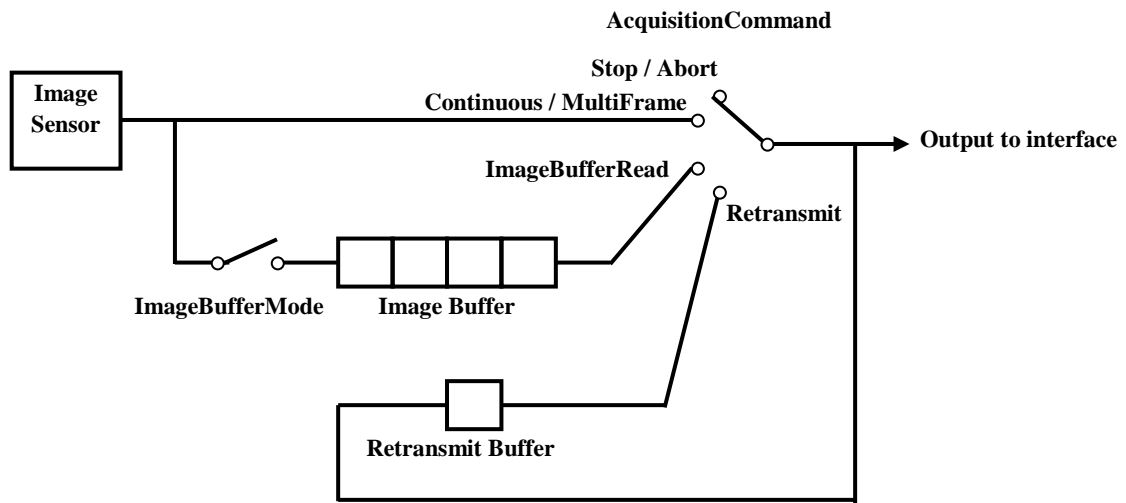


Figure 11 – Block diagram of AcquisitionCommand and ImageBufferMode

In the example shown in the figure below, point **A** is a write request to **ImageBufferMode** enabling the image buffer function. Subsequently two images are acquired and stored in the device. Point **B** is a write request to the **ImageBufferRead** command with **AcquisitionFrameCount** set to two. The device then transfers the two images on the bus from the Image Buffer. Point **C** is a write request to the **Retransmit** command. The device then retransmits the last image.

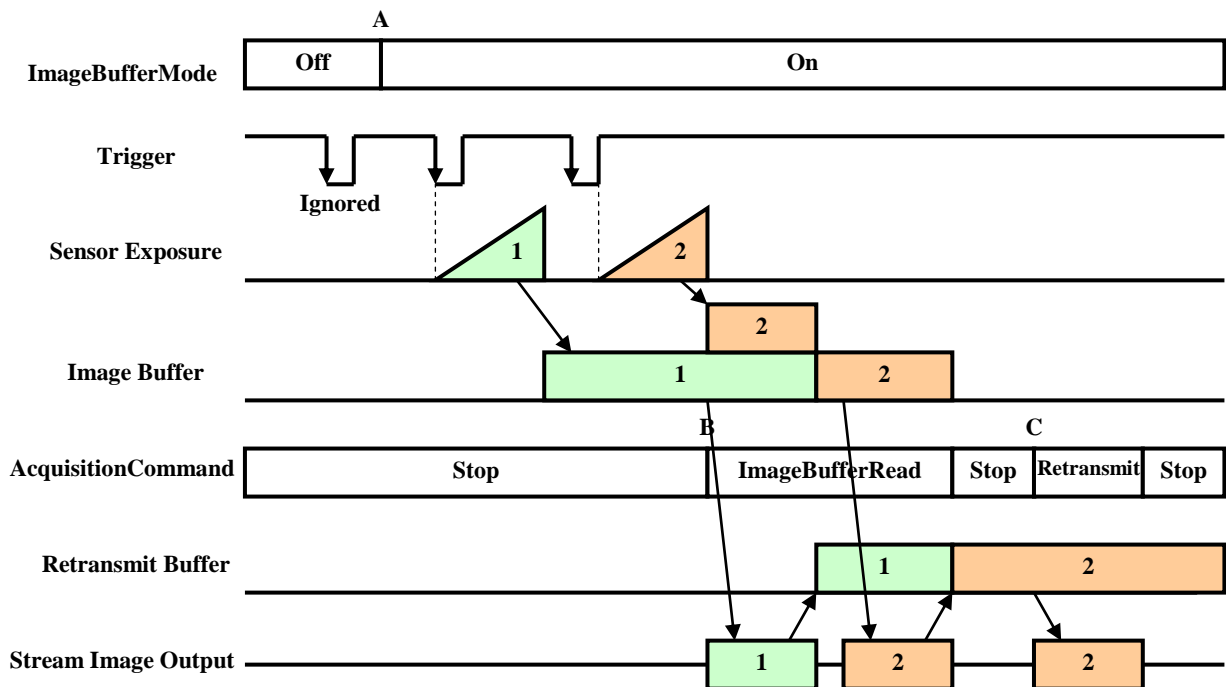


Figure 12 – Example timing of image buffer control

4.4.4 ImageBufferFrameCount

Indicates information about the image buffer. **Max** is the maximum frame number to be stored in Image Buffer. **Value** represents the number of frames in the Image Buffer now.

4.4.5 AcquisitionFrameRate

Handles frame rate of acquisition and transfer.

| | | |
|-------------------------------|-----------|--|
| Unit of Absolute Value | | fps (Frame Per Second) |
| Reference point | | - |
| Recommended value at | NoSpecify | Max (ExposureTime is not limited.) |

[R-57cd]

When **Control** is **Manual**, device shall limit **Max** in **ExposureTime** depending on **Value** in this **FeatureCSR**.

[O-58ch]

Host should check **Max** in **ExposureTime** when this **FeatureCSR** is changed.

[CO-59cd]

If **Value** in **ExposureTime** is out of range, device should change it to **Max** automatically.

[R-60cd]

Device shall update **Max** and **Min** if **FeatureCSRs** in **CategoryBlock2 (ImageFormatControl)** are changed.

[O-61ch]

Host should check **Max** and **Min** after these **FeatureCSRs** are changed.

[O-62cd]

When **Control** is **NoSpecify**, **Value** should indicate available frame rate which depends on other **FeatureCSRs** including **ExposureTime**.

[CR-63cd]

If requirement of **[O-62cd]** is impossible, device SHALL turn **Readable** to **0**.

4.4.6 AcquisitionFrameInterval

It is defined as the reciprocal of **AcquisitionFrameRate**.

| | | |
|-------------------------------|-----------|--|
| Unit of Absolute Value | | second |
| Reference point | | - |
| Recommended value at | NoSpecify | Min (ExposureTime is not limited.) |

4.5 CategoryBlock4 (LuminanceControl)

| Offset | Name | Field | Bit | Description | | |
|------------------|------------------|--------------------------------|----------|-------------------|---------|----------|
| +0x000 | Header | CategoryBlockNumber | [31..24] | = 4 | | |
| | | SizeOfCategoryBlock | [23..0] | = 0x000E0 | | |
| +0x004~ 0x01C | | - | - | reserved | | |
| +0x020 | Exposure Time | Implemented | [31] | | | |
| | | Active | [30] | | | |
| | | - | [29..27] | reserved | | |
| | | UserSetLoadable | [26] | | | |
| | | Writable | [25] | | | |
| | | Readable | [24] | | | |
| | | ValueType | [23..16] | =0x30 (Integer32) | | |
| | | - | [15..5] | reserved | | |
| | | DefaultInq | [4] | | | |
| | | AutoOnceInq | [3] | | | |
| | | AutoInq | [2] | | | |
| | | ManualInq | [1] | | | |
| | | NoSpecifyInq | [0] | | | |
| | | +0x024 | | OffsetForExpanded | [31..0] | |
| | | +0x028 | | - | [31..4] | reserved |
| +0x02C | | Control | [3..0] | | | |
| +0x02C | | Mult | [31..0] | | | |
| +0x030 | | Div | [31..0] | | | |
| +0x034 | | Min | [31..0] | | | |
| +0x038 | | Max | [31..0] | | | |
| +0x03C | | Value | [31..0] | | | |
| +0x040~ 0x05C | BlackLevel | Same Structure as ExposureTime | | | | |
| +0x060~ 0x07C | Gain | Same Structure as ExposureTime | | | | |
| +0x080~ 0x09C | Gamma | Same Structure as ExposureTime | | | | |
| +0x0A0~ 0x0BC | Sharpness | Same Structure as ExposureTime | | | | |
| +0x0C0~ 0x0DC | ALCLevel | Same Structure as ExposureTime | | | | |

Table 33 – CategoryBlock4 (LuminanceControl)

4.5.1 ExposureTime

Integration time of the incoming light.

| | | |
|-------------------------------|-----------------|---|
| Unit of Absolute Value | | second |
| Reference point | | - |
| Behavior of Control | AutoOnce / Auto | Proper value device calculated with ALCLevel |
| | NoSpecify | Maximum value to keep frame rate |

4.5.2 BlackLevel

Black level adjustment of the image.

| | | |
|-------------------------------|-----------------|---|
| Unit of Absolute Value | | % |
| Reference point | | 0 |
| Recommended value at | AutoOnce / Auto | Proper value device calculated with ALCLevel |
| | NoSpecify | Factory setting value |

4.5.3 Gain

Gain control for image.

| | | |
|-------------------------------|-----------------|---|
| Unit of Absolute Value | | dB |
| Reference point | | 0 |
| Recommended value at | AutoOnce / Auto | Proper value device calculated with ALCLevel |
| | NoSpecify | Factory setting value |

4.5.4 Gamma

Define the function between incoming light level and output picture level.

$$Y = X^{\text{Gamma}} \quad Y : \text{output picture level, } X : \text{incoming light level}$$

| | | |
|-------------------------------|-----------------|--------------------------------|
| Unit of Absolute Value | | power-law |
| Reference point | | 1.0 |
| Recommended value at | AutoOnce / Auto | Proper value device calculated |
| | NoSpecify | Factory setting value |

4.5.5 Sharpness

Sharpness of the image.

| | | |
|-------------------------------|-----------------|--------------------------------|
| Unit of Absolute Value | | - |
| Reference point | | 0 (as disable) |
| Recommended value at | AutoOnce / Auto | Proper value device calculated |
| | NoSpecify | Factory setting value |

4.5.6 ALCLevel

Target level for Auto Luminance Control.

When either **ExposureTime**, **BlackLevel** or **Gain** is set to **Auto** or **AutoOnce**, device handles these features automatically in order to get appropriate level of image.

| | | |
|-------------------------------|-----------------|--------------------------------|
| Unit of Absolute Value | | EV (Exposure Value) |
| Reference point | | 0 |
| Recommended value at | AutoOnce / Auto | Proper value device calculated |
| | NoSpecify | Factory setting value |

4.6 CategoryBlock5 (ChromaControl)

| Offset | Name | Field | Bit | Description |
|------------------|----------------------|-----------------------|----------|--------------------|
| +0x000 | Header | CategoryBlockNumber | [31..24] | = 5 |
| | | SizeOfCategoryBlock | [23..0] | = 0x000100 |
| +0x004~ 0x01C | | - | - | reserved |
| +0x020 | Hue | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | |
| | | Readable | [24] | |
| | | ValueType | [23..16] | = 0x30 (Integer32) |
| | | - | [15..5] | reserved |
| | | DefaultInq | [4] | |
| | | AutoOnceInq | [3] | |
| | | AutoInq | [2] | |
| | | ManualInq | [1] | |
| | | NoSpecifyInq | [0] | |
| | | +0x024 | | OffsetForExpanded |
| +0x028 | | - | [31..4] | reserved |
| | | Control | [3..0] | |
| +0x02C | | Mult | [31..0] | |
| +0x030 | | Div | [31..0] | |
| +0x034 | | Min | [31..0] | |
| +0x038 | | Max | [31..0] | |
| +0x03C | | Value | [31..0] | |
| +0x040~ 0x05C | Saturation | Same Structure as Hue | | |
| +0x060~ 0x07C | WhiteBalance R | Same Structure as Hue | | |
| +0x080~ 0x09C | WhiteBalance B | Same Structure as Hue | | |
| +0x0A0~ 0x0BC | WhiteBalance U | Same Structure as Hue | | |
| +0x0C0~ 0x0DC | WhiteBalance V | Same Structure as Hue | | |
| +0x0E0~ 0x0FC | Color Temperature | Same Structure as Hue | | |

Table 34 – CategoryBlock5 (ChromaControl)

4.6.1 Hue

Color phase of the image.

| | | |
|-------------------------------|-----------------|--------------------------------|
| Unit of Absolute Value | | degree |
| Reference point | | 0 |
| Recommended value at | AutoOnce / Auto | Proper value device calculated |
| | NoSpecify | Factory setting value |

4.6.2 Saturation

Color saturation of the image.

| | | |
|-------------------------------|-----------------|--------------------------------|
| Unit of Absolute Value | | % |
| Reference point | | 0 |
| Recommended value at | AutoOnce / Auto | Proper value device calculated |
| | NoSpecify | Factory setting value |

4.6.3 WhiteBalanceR / WhiteBalanceB

Adjustment of the white color of the picture. **WhiteBalanceR** handles red plane, and **WhiteBalanceB** is Blue. **Value** is relative setting from green plane.

[CR-64cd]

If the device cannot be set to **Auto** / **AutoOnce** separately, the device shall refer **Control** fields to each other.

| | | |
|-------------------------------|-----------------|--------------------------------|
| Unit of Absolute Value | | dB |
| Reference point | | 0 |
| Recommended value at | AutoOnce / Auto | Proper value device calculated |
| | NoSpecify | Factory setting value |

4.6.4 WhiteBalanceU / WhiteBalanceV

Adjustment of the white color of the picture. **WhiteBalanceU** handles chrominance red, and **WhiteBalanceV** is chrominance Blue. **Value** is offset level.

[CR-65cd]

If the device cannot be set to **Auto** / **AutoOnce** separately, the device shall refer **Control** fields to each other.

| | | |
|-------------------------------|-----------------|--------------------------------|
| Unit of Absolute Value | | % |
| Reference point | | 0 |
| Recommended value at | AutoOnce / Auto | Proper value device calculated |
| | NoSpecify | Factory setting value |

4.6.5 ColorTemperature

Adjustment of the color temperature of the picture.

| | | |
|-------------------------------|-----------------|--------------------------------|
| Unit of Absolute Value | | Kelvin |
| Reference point | | 0 |
| Recommended value at | AutoOnce / Auto | Proper value device calculated |
| | NoSpecify | Factory setting value |

4.7 CategoryBlock6 (LUTControl)

| Offset | Name | Field | Bit | Description |
|------------------|-------------------------|---------------------|----------|----------------------|
| +0x000 | Header | CategoryBlockNumber | [31..24] | = 6 |
| | | SizeOfCategoryBlock | [23..0] | =0x000080 |
| +0x004~ 0x01C | | - | - | reserved |
| +0x020 | LUT Enable | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | |
| | | Readable | [24] | |
| | | ValueType | [23..16] | = 0x03 (Enumeration) |
| | | - | [15..5] | reserved |
| | | DefaultInq | [4] | |
| | | AutoOnceInq | [3] | not-used |
| | | AutoInq | [2] | not-used |
| | | ManualInq | [1] | |
| | | NoSpecifyInq | [0] | not-used |
| | | +0x024 | | OffsetForExpanded |
| +0x028 | | - | [31..4] | reserved |
| | | Control | [3..0] | |
| +0x02C~ 0x038 | ListOfElements | | [127] | VendorSpecific7 |
| | | ... | ... | ... |
| | | | [121] | VendorSpecific1 |
| | | | [120] | VendorSpecific0 |
| | | | [119..2] | not-used |
| | | | [1] | On |
| | [0] | Off | | |
| +0x03C | | Value | [31..0] | |
| +0x040 | LUT Bank Selector | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | |
| | | Readable | [24] | |
| | | ValueType | [23..16] | = 0x03 (Enumeration) |
| | | - | [15..5] | reserved |
| | | DefaultInq | [4] | |
| | | AutoOnceInq | [3] | not-used |
| | | AutoInq | [2] | not-used |
| | | ManualInq | [1] | |
| | | NoSpecifyInq | [0] | not-used |
| | | +0x044 | | OffsetForExpanded |
| +0x048 | | - | [31..4] | reserved |
| | | Control | [3..0] | |

Table 35 – CategoryBlock6 (LUTControl)

| Offset | Name | Field | Bit | Description |
|------------------|-------------------------|-------------------|----------|----------------------|
| +0x04C~ 0x058 | LUT Bank Selector | ListOfElements | [127] | VendorSpecific7 |
| | | | ... | ... |
| | | | [121] | VendorSpecific1 |
| | | | [120] | VendorSpecific0 |
| | | | [119..2] | not-used |
| | | | [31] | Bank31 |
| | | | ... | ... |
| | | | [1] | Bank1 |
| [0] | Bank0 | | | |
| +0x05C | | Value | [31..0] | |
| +0x060 | LUTValue All | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | |
| | | Readable | [24] | |
| | | ValueType | [23..16] | = 0x03 (Enumeration) |
| | | ControlInq | [15..0] | not-used |
| +0x064 | | OffsetForExpanded | [31..0] | |
| +0x068 | | Control | [31..0] | = 0x01 (Manual) |
| +0x06C~ 0x078 | | ListOfElements | [127] | VendorSpecific7 |
| | | | ... | ... |
| | | | [121] | VendorSpecific1 |
| | | | [120] | VendorSpecific0 |
| | | | [119..2] | not-used |
| | | | [31] | Preset31 |
| | | | ... | ... |
| | | | [1] | Preset1 |
| [0] | Preset0 | | | |
| +0x07C | | Value | [31..0] | |

Table 35 – CategoryBlock6 (LUTControl) (Contd.)

4.7.1 LUTEnable

Handles LUT function.

On : Enable LUT function. LUT number is selected by **LUTBankSelector** .

Off : Disable LUT function.

VendorSpecific0~7 :Camera vendor specific pattern

4.7.2 LUTBankSelector

Selects the bank of LUT.

4.7.3 LUTValueAll

This feature is implemented as **ExpandedCSR** pointed by **OffsetForExpanded** when LUT value is either readable or writable. It is defined by **ArrayOfInteger32** with 2-dimension.

[CR-66c]

LUT Value table shall be placed with structure as following table.

| Offset | Name | Field | Bit | Description |
|--------------|-----------------|-------------------|----------|------------------------------------|
| +0x000 | LUTValue All | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | |
| | | Readable | [24] | |
| | | ValueType | [23..16] | = 0xB0 (ArrayOfInteger32) |
| | | - | [15..5] | reserved |
| | | DefaultInq | [4] | |
| | | AutoOnceInq | [3] | |
| | | AutoInq | [2] | |
| | | ManualInq | [1] | |
| | | NoSpecifyInq | [0] | |
| | | +0x004 | | OffsetForExpanded |
| +0x008 | | - | [31..4] | reserved |
| | | Control | [3..0] | |
| +0x00C | | MoreDimension | [31] | = 1 |
| | | NumberOfChannels | [30..0] | Number of channels (Cn) |
| +0x010 | | MoreDimension | [31] | = 0 |
| | | NumberOfElements | [30..0] | Number of elements per a bank (En) |
| +0x014 | | Mult | [31..0] | |
| +0x018 | | Div | [31..0] | |
| +0x01C | | Min | [31..0] | |
| +0x020 | | Max | [31..0] | |
| +0x024 | | Value[0][0] | [31..0] | |
| +0x028 | | Value[0][1] | [31..0] | |
| ... | | | [31..0] | |
| +0xXXX | | Value[0][En-1] | [31..0] | |
| +0xXXX +4 | | Value[1][0] | [31..0] | |
| ... | | ... | | |
| +0xYYY | | Value[Cn-1][En-1] | [31..0] | |

Table 36 – LUTValueAll in ExpandedCategoryBlock (chaining CSR)

4.8 CategoryBlock7 (TriggerControl)

| Offset | Name | Field | Bit | Description |
|------------------|-----------------|---------------------|----------|----------------------|
| +0x000 | Header | CategoryBlockNumber | [31..24] | = 7 |
| | | SizeOfCategoryBlock | [23..0] | = 0x0000E0 |
| +0x004~ 0x01C | | - | - | reserved |
| +0x020 | Trigger Mode | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | |
| | | Readable | [24] | |
| | | ValueType | [23..16] | = 0x03 (Enumeration) |
| | | - | [15..5] | reserved |
| | | DefaultInq | [4] | |
| | | AutoOnceInq | [3] | not-used |
| | | AutoInq | [2] | not-used |
| | | ManualInq | [1] | |
| | | NoSpecifyInq | [0] | not-used |
| | | +0x024 | | OffsetForExpanded |
| +0x028 | | - | [31..4] | reserved |
| | | Control | [3..0] | |
| +0x02C~ 0x038 | ListofElements | | [127] | VendorSpecific7 |
| | | ... | ... | ... |
| | | | [121] | VendorSpecific1 |
| | | | [120] | VendorSpecific0 |
| | | | [119..2] | not-used |
| | | | [1] | On |
| | | [0] | Off | |
| +0x03C | | Value | [31..0] | |

Table 37 – CategoryBlock7 (TriggerControl)

| Offset | Name | Field | Bit | Description |
|------------------|------------------|------------------|----------|----------------------|
| +0x040 | Trigger Sequence | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | |
| | | Readable | [24] | |
| | | ValueType | [23..16] | = 0x03 (Enumeration) |
| | | - | [15..5] | reserved |
| | | DefaultInq | [4] | |
| | | AutoOnceInq | [3] | not-used |
| | | AutoInq | [2] | not-used |
| | | ManualInq | [1] | |
| | | NoSpecifyInq | [0] | not-used |
| | | +0x044 | | OffsetForExpanded |
| +0x048 | | - | [31..4] | reserved |
| | | Control | [3..0] | |
| +0x04C~ 0x058 | ListOfElements | | [127] | VendorSpecific7 |
| | | ... | ... | ... |
| | | | [121] | VendorSpecific1 |
| | | | [120] | VendorSpecific0 |
| | | | [119..7] | not-used |
| | | | [6] | TriggerSequence6 |
| | | | [5] | TriggerSequence5 |
| | | | [4] | TriggerSequence4 |
| | | | [3] | TriggerSequence3 |
| | | | [2] | TriggerSequence2 |
| | [1] | TriggerSequence1 | | |
| | [0] | TriggerSequence0 | | |
| +0x05C | | Value | [31..0] | |

Table 37 – CategoryBlock7 (TriggerControl) (Contd.)

| Offset | Name | Field | Bit | Description |
|------------------|------------------------------|-------------------|-----------|----------------------|
| +0x060 | Trigger Source | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | |
| | | Readable | [24] | |
| | | ValueType | [23..16] | = 0x03 (Enumeration) |
| | | - | [15..5] | Reserved |
| | | DefaultInq | [4] | |
| | | AutoOnceInq | [3] | not-used |
| | | AutoInq | [2] | not-used |
| | | ManualInq | [1] | |
| | | NoSpecifyInq | [0] | not-used |
| +0x064 | | OffsetForExpanded | [31..0] | |
| +0x068 | | - | [31..4] | Reserved |
| | | Control | [3..0] | |
| +0x06C~ 0x078 | ListofElements | | [127] | VendorSpecific7 |
| | | | ... | ... |
| | | | [121] | VendorSpecific1 |
| | | | [120] | VendorSpecific0 |
| | | | [119..65] | not-used |
| | | | [64] | SoftwareTrigger |
| | | | [63..32] | not-used |
| | | | [31] | IOLine31 |
| | | | ... | ... |
| | | | [1] | IOLine1 |
| | [0] | IOLine0 | | |
| +0x07C | | Value | [31..0] | |
| +0x080 | Trigger Additional Parameter | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | |
| | | Readable | [24] | |
| | | ValueType | [23..16] | = 0x30 (Integer32) |
| | | - | [15..5] | reserved |
| | | DefaultInq | [4] | |
| | | AutoOnceInq | [3] | not-used |
| | | AutoInq | [2] | not-used |
| | | ManualInq | [1] | |
| | | NoSpecifyInq | [0] | not-used |
| +0x084 | | OffsetForExpanded | [31..0] | |
| +0x088 | | - | [31..4] | reserved |
| | | Control | [3..0] | |
| +0x08C | | Mult | [31..0] | |
| +0x090 | | Div | [31..0] | |
| +0x094 | | Min | [31..0] | |
| +0x098 | | Max | [31..0] | |
| +0x09C | | Value | [31..0] | |

Table 37 – CategoryBlock7 (TriggerControl) (Contd.)

| Offset | Name | Field | Bit | Description |
|------------------|------------------|-----------------|----------|----------------------|
| +0x0A0 | Trigger Delay | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | |
| | | Readable | [24] | |
| | | ValueType | [23..16] | = 0x30 (Integer32) |
| | | - | [15..5] | reserved |
| | | DefaultInq | [4] | |
| | | AutoOnceInq | [3] | |
| | | AutoInq | [2] | |
| | | ManualInq | [1] | |
| | | NoSpecifyInq | [0] | |
| | | +0x0A4 | | OffsetForExpanded |
| +0x0A8 | | - | [31..4] | reserved |
| | | Control | [3..0] | |
| +0x0AC | | Mult | [31..0] | |
| +0x0B0 | | Div | [31..0] | |
| +0x0B4 | | Min | [31..0] | |
| +0x0B8 | | Max | [31..0] | |
| +0x0BC | | Value | [31..0] | |
| +0x0C0 | Software Trigger | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | |
| | | Readable | [24] | |
| | | ValueType | [23..16] | = 0x03 (Enumeration) |
| | | - | [15..5] | reserved |
| | | DefaultInq | [4] | |
| | | AutoOnceInq | [3] | not-used |
| | | AutoInq | [2] | not-used |
| | | ManualInq | [1] | |
| | | NoSpecifyInq | [0] | not-used |
| | | +0x0C4 | | OffsetForExpanded |
| +0x0C8 | | - | [31..4] | reserved |
| | | Control | [3..0] | |
| +0x0CC~ 0x0D8 | ListofElements | | [127] | VendorSpecific7 |
| | | ... | ... | |
| | | | [121] | VendorSpecific1 |
| | | | [120] | VendorSpecific0 |
| | | | [119..9] | not-used |
| | | | [8] | Impulse |
| | | | [7..2] | not-used |
| | | | [1] | Active |
| | [0] | Inactive | | |
| +0x0DC | | Value | [31..0] | |

Table 37 – CategoryBlock7 (TriggerControl) (Contd.)

4.8.1 TriggerMode

Selects the trigger mode for acquiring image.

Off : Acquiring image by normal operation.

On : Acquiring image by external trigger mode.

VendorSpecific0~7 :Camera vendor specific pattern

4.8.2 TriggerSequence

Selects the trigger sequence.

Sequence0 : External Edge mode

Device starts integration of the incoming light from external trigger input falling edge. Integration time is described in **ExposureTime**. **TriggerAdditionalParameter** is not used.

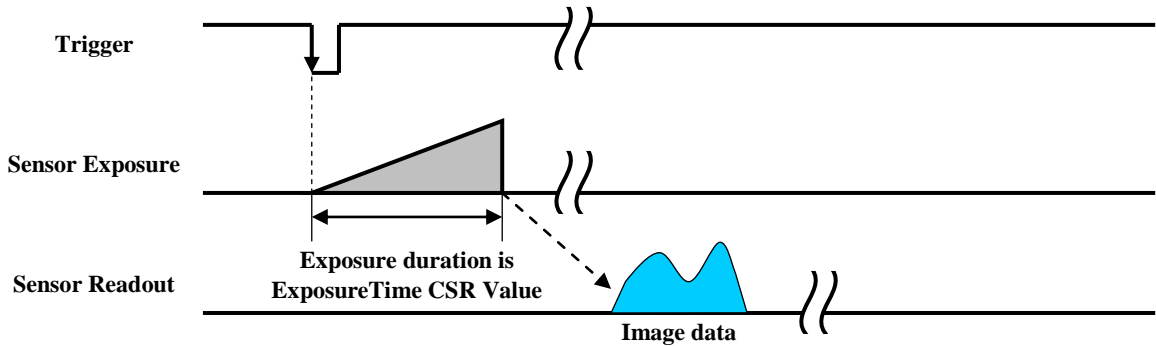


Figure 13 – Trigger Sequence0

Sequence1 : External level mode

Device starts integration of the incoming light from external trigger input falling edge. Integration time is equal to low state time of the external trigger input. **ExposureTime** and **TriggerAdditionalParameter** is not used.

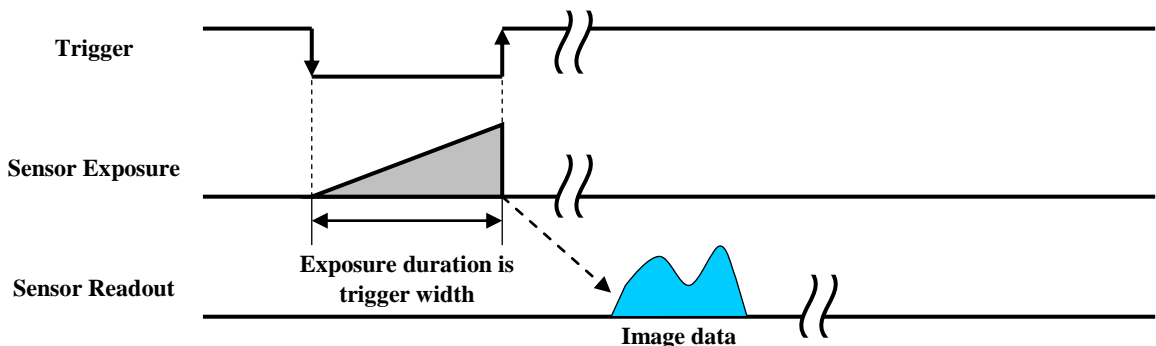


Figure 14 – Trigger Sequence1

Sequence2 : External event mode

Device starts integration of incoming light from first external trigger input falling edge. At the N-th (define by **TriggerAdditionalParameter**) external trigger input falling edge, integration will be stopped.

[CCR-67c]

TriggerAdditionalParameter is required and shall be two or more. ($N \geq 2$)

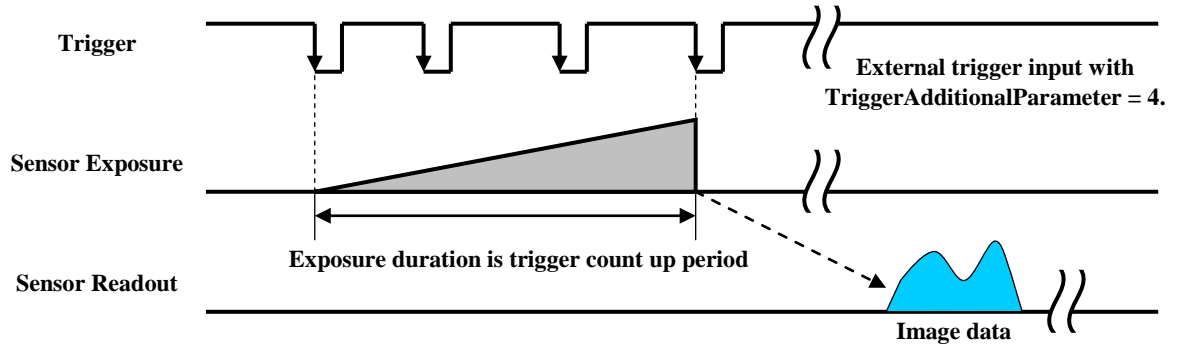


Figure 15 – Trigger Sequence2

Sequence3 : Frame Interval mode

This is an internal trigger sequence. Device will issue trigger internally and internal trigger cycle time is **TriggerAdditionalParameter** times of the minimum frame interval. Integration time of incoming light is described in **ExposureTime**.

[CR-68cd]

TriggerAdditionalParameter is required and shall be one or more. ($N \geq 1$)

Internal trigger cycle time is **TriggerAdditionalParameter** value x T_f
 (T_f : minimum frame interval)

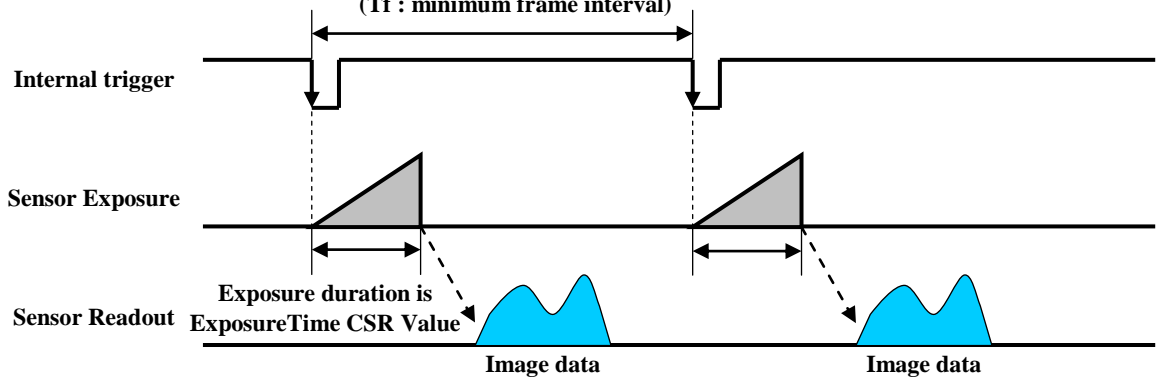


Figure 16 – Trigger Sequence3

Sequence4 : Multiple Shutter Preset mode

Device starts integration of incoming light from first external trigger input falling edge and exposes incoming light at **ExposureTime**. Repeat this sequence the N-th (define by **TriggerAdditionalParameter**) external trigger input falling edge then finish integration.

[CR-69cd]

TriggerAdditionalParameter is required and shall be one or more. (N >= 1)

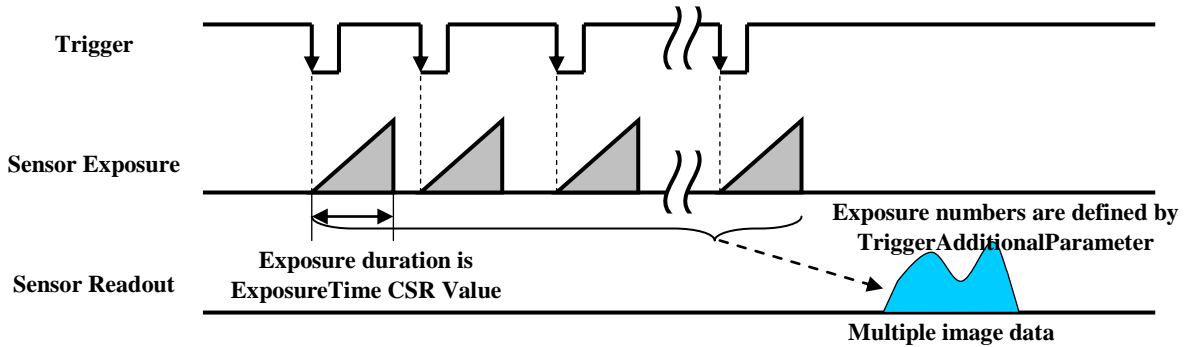


Figure 17 – Trigger Sequence4

Sequence5 : Multiple Shutter Pulse Width mode

Device starts integration of incoming light from first external trigger input falling edge and exposes incoming light until trigger is inactive. Repeat this sequence the N-th (defined by **TriggerAdditionalParameter**) external trigger input falling edge then finish integration.

[CR-70cd]

TriggerAdditionalParameter is required and shall be one or more. (N >= 1)

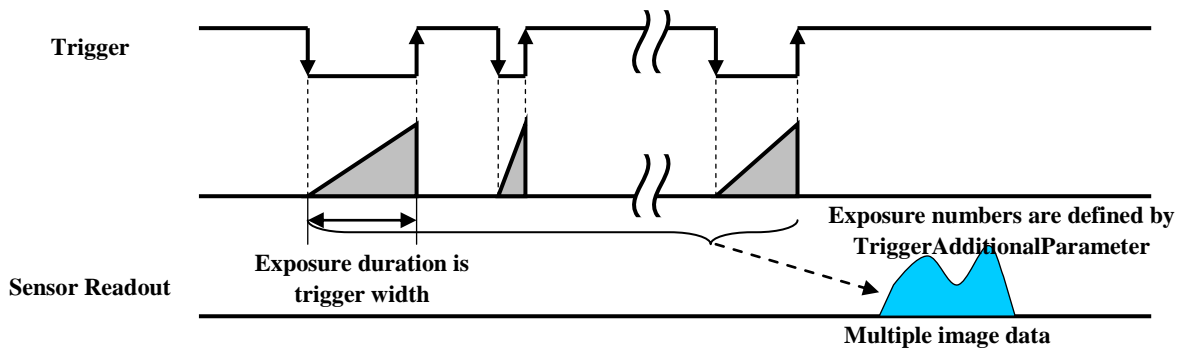


Figure 18 – Trigger Sequence5

Sequence6 : Frame Burst mode

Device starts integration of incoming light from external trigger input falling edge. Integration time is described in **ExposureTime**. After the 1st exposure device starts integration again as soon as possible. Repeat this sequence the N-th (define by **TriggerAdditionalParameter**) external trigger input falling edge then finish integration.

[CR-71cd]

TriggerAdditionalParameter is required and shall be one or more. (N >= 1)

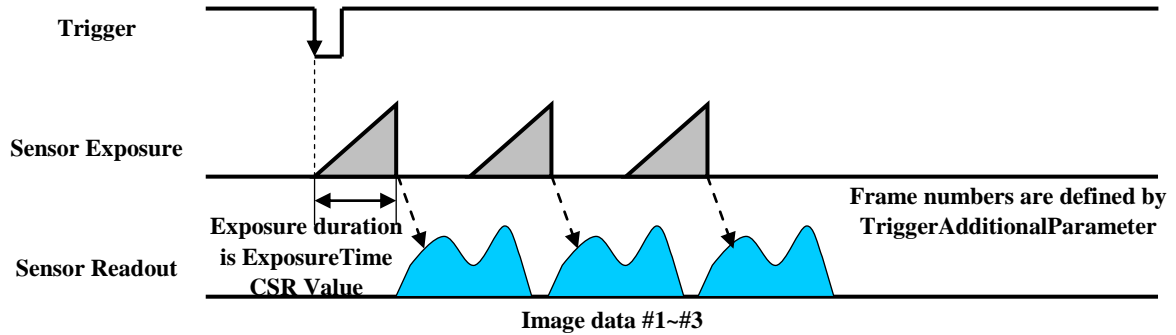


Figure 19 – Trigger Sequence6

VendorSpecific0~7 :Camera vendor specific pattern

4.8.3 TriggerSource

Selects trigger source at the external trigger mode. External trigger is always active-lo. Device and host may use invert signal with handling **IOLineInverterAll** in **CategoryBlock9 (DigitalIOControl)**.

4.8.4 TriggerAdditionalParameter

Trigger parameter at the external trigger mode if needed (It depends on **TriggerSequence**).

| | | |
|-------------------------------|-----------------|--------------------------------|
| Unit of Absolute Value | | - |
| Reference point | | - |
| Recommended value at | AutoOnce / Auto | Proper value device calculated |
| | NoSpecify | Factory setting value |

4.8.5 TriggerDelay

Add internal delay of trigger signal.

| | | |
|-------------------------------|-----------------|--------------------------------|
| Unit of Absolute Value | | second |
| Reference point | | 0 |
| Recommended value at | AutoOnce / Auto | Proper value device calculated |
| | NoSpecify | Factory setting value |

4.8.6 SoftwareTrigger

Trigger input at the software trigger mode.

Inactive : Set software trigger to inactive

Active : Set software trigger to active.

Impulse: Input the impulse for software trigger.

VendorSpecific0~7 :Camera vendor specific pattern

[R-72cd]

After the impulse function is finished, device shall turn to **Inactive** automatically.

4.9 CategoryBlock8 (UserSetControl)

| Offset | Name | Field | Bit | Description |
|------------------|--------------------|---------------------|-----------|----------------------|
| +0x000 | Header | CategoryBlockNumber | [31..24] | = 8 |
| | | SizeOfCategoryBlock | [23..0] | = 0x0000C0 |
| +0x004~ 0x01C | | - | - | reserved |
| +0x020 | Standard Format | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | |
| | | Readable | [24] | |
| | | ValueType | [23..16] | = 0x03 (Enumeration) |
| | | ControlInq | [15..0] | not-used |
| +0x024 | | OffsetForExpanded | [31..0] | |
| +0x028 | | Control | [31..0] | = 0x01 (Manual) |
| +0x02C~ 0x038 | ListofElements | | [127..43] | not-used |
| | | | [42] | HD1080p_YUV422Packed |
| | | | [41] | HD1080p_RGB8Packed |
| | | | [40] | HD1080p_Mono8 |
| | | | [39..35] | not-used |
| | | | [34] | UXGA_YUV422Packed |
| | | | [33] | UXGA_RGB8Packed |
| | | | [32] | UXGA_Mono8 |
| | | | [31..27] | not-used |
| | | | [26] | SXGA_YUV422Packed |
| | | | [25] | SXGA_RGB8Packed |
| | | | [24] | SXGA_Mono8 |
| | | | [23..19] | not-used |
| | | | [18] | HD720p_YUV422Packed |
| | | | [17] | HD720p_RGB8Packed |
| | | | [16] | HD720p_Mono8 |
| | | | [15..11] | not-used |
| | | | [10] | XGA_YUV422Packed |
| | | | [9] | XGA_RGB8Packed |
| | | | [8] | XGA_Mono8 |
| | [7..3] | not-used | | |
| | [2] | VGA_YUV422Packed | | |
| | [1] | VGA_RGB8Packed | | |
| | [0] | VGA_Mono8 | | |
| +0x03C | | Value | [31..0] | |

Table 38 – CategoryBlock8 (UserSetControl)

| Offset | Name | Field | Bit | Description |
|------------------|--------------------|-------------------|-----------|----------------------|
| +0x040 | Standard FrameRate | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | |
| | | Readable | [24] | |
| | | ValueType | [23..16] | = 0x03 (Enumeration) |
| | | ControlInq | [15..0] | not-used |
| +0x044 | | OffsetForExpanded | [31..0] | |
| +0x048 | | Control | [31..0] | = 0x01 (Manual) |
| +0x04C~ 0x058 | ListofElements | | [127..8] | not-used |
| | | | [7] | 240 fps |
| | | | [6] | 120 fps |
| | | | [5] | 60 fps |
| | | | [4] | 30 fps |
| | | | [3] | 15 fps |
| | | | [2] | 7.5 fps |
| | | | [1] | 3.75 fps |
| | | | [0] | 1.875 fps |
| +0x05C | | Value | [31..0] | |
| +0x060 | UserSet Selector | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | |
| | | Readable | [24] | |
| | | ValueType | [23..16] | = 0x03 (Enumeration) |
| | | ControlInq | [15..0] | not-used |
| +0x064 | | OffsetForExpanded | [31..0] | |
| +0x068 | | Control | [31..0] | = 0x01 (Manual) |
| +0x06C~ 0x078 | ListofElements | | [127] | VendorSpecific7 |
| | | | ... | ... |
| | | | [121] | VendorSpecific1 |
| | | | [120] | VendorSpecific0 |
| | | | [119..32] | not-used |
| | | | [31] | UserSet31 |
| | | | ... | ... |
| | | | [1] | UserSet0 |
| | [0] | Default | | |
| +0x07C | | Value | [31..0] | |

Table 38 – CategoryBlock8 (UserSetControl) (Contd.)

| Offset | Name | Field | Bit | Description |
|------------------|-----------------|-------------------|-----------|----------------------|
| +0x080 | UserSet Command | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | |
| | | Readable | [24] | |
| | | ValueType | [23..16] | = 0x03 (Enumeration) |
| | | ControlInq | [15..0] | not-used |
| +0x084 | | OffsetForExpanded | [31..0] | |
| +0x088 | | Control | [31..0] | = 0x01 (Manual) |
| +0x08C~ 0x098 | ListofElements | | [127] | VendorSpecific7 |
| | | ... | ... | ... |
| | | | [121] | VendorSpecific1 |
| | | | [120] | VendorSpecific0 |
| | | | [119..10] | not-used |
| | | | [9] | Save |
| | | | [8] | Load |
| | | | [7..1] | not-used |
| | [0] | Done | | |
| +0x09C | | Value | [31..0] | |
| +0x0A0 | UserSet Default | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | |
| | | Readable | [24] | |
| | | ValueType | [23..16] | = 0x03 (Enumeration) |
| | | ControlInq | [15..0] | not-used |
| +0x0A4 | | OffsetForExpanded | [31..0] | |
| +0x0A8 | | Control | [31..0] | = 0x01 (Manual) |
| +0x0AC~ 0x0B8 | ListofElements | | [127] | VendorSpecific7 |
| | | ... | ... | ... |
| | | | [121] | VendorSpecific1 |
| | | | [120] | VendorSpecific0 |
| | | | [119..32] | not-used |
| | | | [31] | UserSet31 |
| | | ... | ... | ... |
| | | | [1] | UserSet0 |
| | [0] | Default | | |
| +0x0BC | | Value | [31..0] | |

Table 38 – CategoryBlock8 (UserSetControl) (Contd.)

4.9.1 StandardFormat / StandardFrameRate

Support settings for well-known image formats. Host MAY receive image format with same setting. The device may limit **StandardFrameRate** by **StandardFormat**. Device supporting these **FeatureCSRs** has the table of register set for each image format.

[R-73cd]

If host writes these **FeatureCSRs**, device shall apply register set to corresponding **FeatureCSRs** (**ImageSize**, **PixelCoding**, **PixelSize**, **AcquisitionFrameRate**, **AcquisitionFrameInterval** and **FeatureCSRs in TransportLayerControl**) immediately.

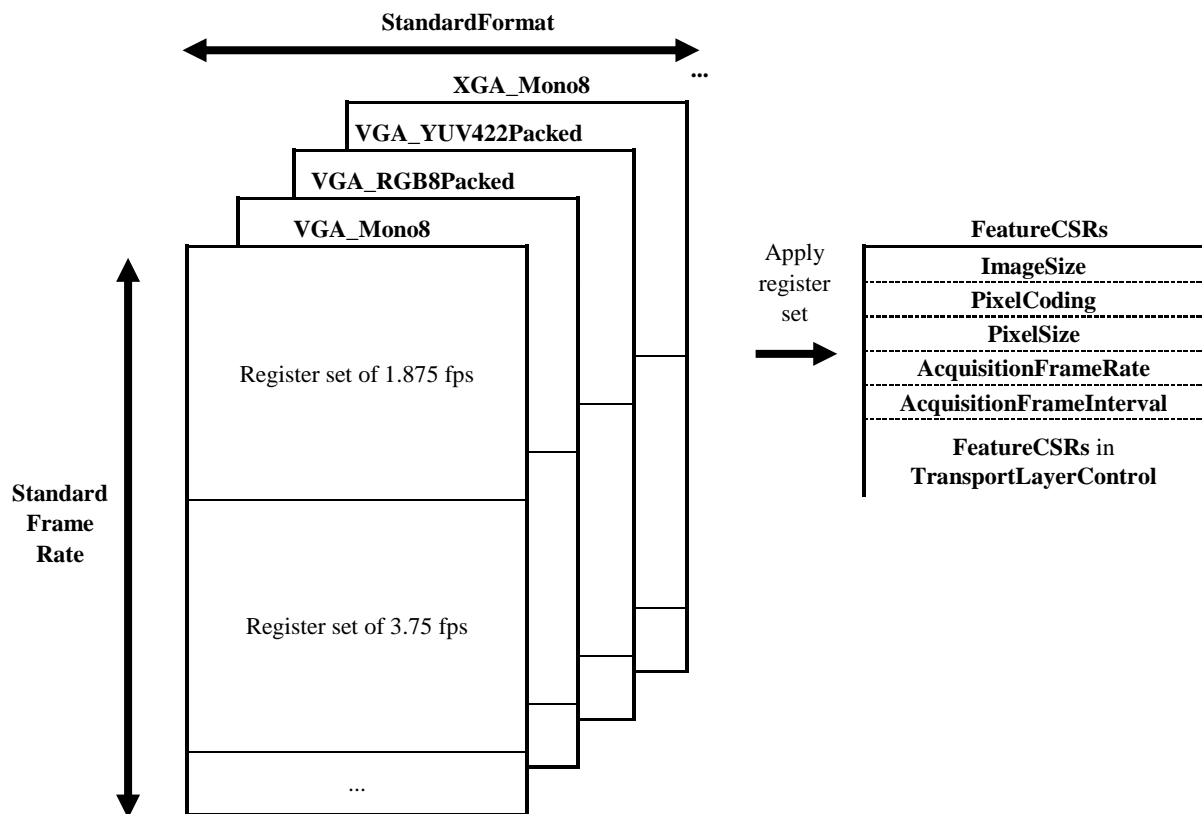


Figure 20 – StandardFormat / StandardFrameRate

Relationship between **ImageSize** and type of **StandardFormat** is as follows.

| Type of StandardFormat | ImageSize | |
|------------------------|-----------|--------|
| | Width | height |
| VGA | 640 | 480 |
| XGA | 1024 | 768 |
| HD720p | 1280 | 720 |
| SXGA | 1280 | 960 |
| UXGA | 1600 | 1200 |
| HD1080p | 1920 | 1080 |

Table 39 –Relationship between ImageSize and type of StandardFormat

Actual values of **AcquisitionFrameRate** and **AcquisitionFrameInterval** depend on the device (**AbsoluteValues** are same as **StandardFrameRate**).

About **FeatureCSRs** in **TransportLayerControl**, please see each specification of transport layer.

4.9.2 UserSetSelector / UserSetCommand

Handle non-volatile memory in the device. **UserSetSelector** selects the target page, and **UserSetCommand** handles execution of memory command.

Default is the factory setting page. **UserSet1, UserSet2...** are user setting pages.

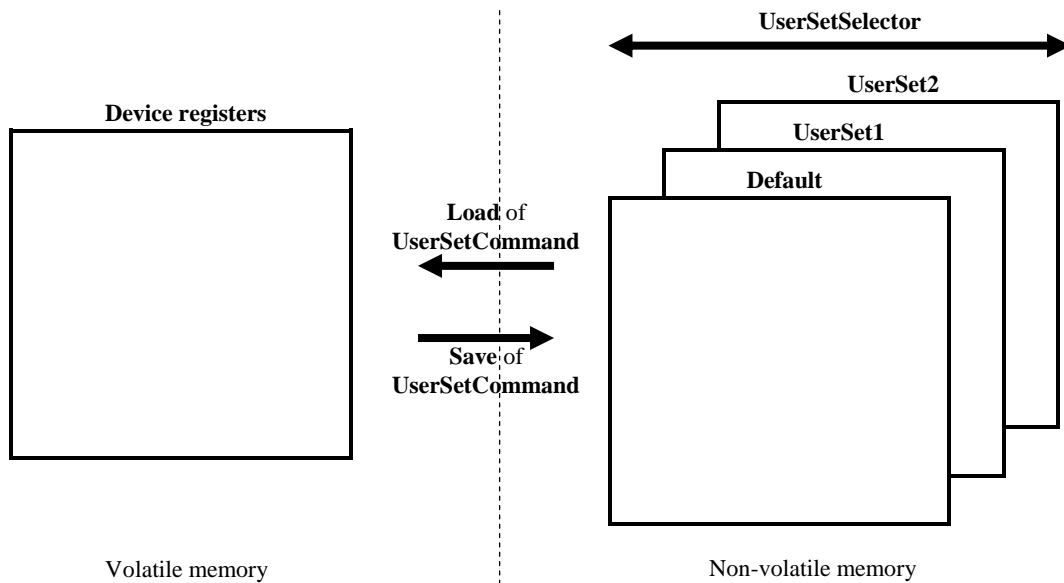


Figure 21 – UserSetSelector / UserSetCommand

4.9.3 UserSetDefault

Handle automatic load function when device is boot or reset.

Default is no load, device will be boot with factory setting registers. **UserSet1, UserSet2...** are recommended setting for auto-load.

4.10 CategoryBlock9 (DigitalIOControl)

| Offset | Name | Field | Bit | Description |
|------------------|-------------------|---------------------|----------|------------------------|
| +0x000 | Header | CategoryBlockNumber | [31..24] | = 9 |
| | | SizeOfCategoryBlock | [23..0] | = 0x0000E0 |
| +0x004~ 0x01C | | - | - | reserved |
| +0x020 | IOLine ModeAll | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | |
| | | Readable | [24] | |
| | | ValueType | [23..16] | = 0x34 (BulkBoolean32) |
| | | - | [15..5] | reserved |
| | | DefaultInq | [4] | |
| | | AutoOnceInq | [3] | not-used |
| | | AutoInq | [2] | not-used |
| | | ManualInq | [1] | |
| | | NoSpecifyInq | [0] | not-used |
| | | +0x024 | | OffsetForExpanded |
| +0x028 | | - | [31..4] | reserved |
| | | Control | [3..0] | |
| +0x02C | | BitWritable | [31..0] | |
| +0x030 | | Value | [31..0] | |
| +0x034~ 0x03C | | - | - | Reserved |

Table 40 – CategoryBlock9 (DigitalIOControl)

| Offset | Name | Field | Bit | Description | | |
|------------------|-----------------------|------------------|----------|------------------------|---------|-----------------|
| +0x040 | IOLine InverterAll | Implemented | [31] | | | |
| | | Active | [30] | | | |
| | | - | [29..27] | reserved | | |
| | | UserSetLoadable | [26] | | | |
| | | Writable | [25] | | | |
| | | Readable | [24] | | | |
| | | ValueType | [23..16] | = 0x34 (BulkBoolean32) | | |
| | | - | [15..5] | reserved | | |
| | | DefaultInq | [4] | | | |
| | | AutoOnceInq | [3] | not-used | | |
| | | AutoInq | [2] | not-used | | |
| | | ManualInq | [1] | | | |
| | | NoSpecifyInq | [0] | not-used | | |
| | | +0x044 | | OffsetForExpanded | [31..0] | |
| +0x048 | | - | [31..4] | reserved | | |
| | | Control | [3..0] | | | |
| +0x04C | | BitWritable | [31..0] | | | |
| +0x050 | | Value | [31..0] | | | |
| +0x054~ 0x05C | | - | - | Reserved | | |
| +0x060 | IOLine StatusAll | Implemented | [31] | | | |
| | | Active | [30] | | | |
| | | - | [29..27] | reserved | | |
| | | UserSetLoadable | [26] | | | |
| | | Writable | [25] | = 0 (read-only) | | |
| | | Readable | [24] | | | |
| | | ValueType | [23..16] | = 0x34 (BulkBoolean32) | | |
| | | ControlInq | [15..0] | not-used | | |
| | | +0x064 | | OffsetForExpanded | [31..0] | |
| | | +0x068 | | Control | [31..0] | = 0x01 (Manual) |
| | | +0x06C | | BitWritable | [31..0] | not-used |
| | | +0x070 | | Value | [31..0] | |
| | | +0x074~ 0x07C | | - | - | Reserved |

Table 40 – CategoryBlock9 (DigitalIOControl) (Contd.)

| Offset | Name | Field | Bit | Description | | |
|------------------|------------------------|------------------|----------------|------------------------|---------|-----------------|
| +0x080 | UserOutput ValueAll | Implemented | [31] | | | |
| | | Active | [30] | | | |
| | | - | [29..27] | reserved | | |
| | | UserSetLoadable | [26] | | | |
| | | Writable | [25] | | | |
| | | Readable | [24] | | | |
| | | ValueType | [23..16] | = 0x34 (BulkBoolean32) | | |
| | | - | [15..5] | reserved | | |
| | | DefaultInq | [4] | | | |
| | | AutoOnceInq | [3] | not-used | | |
| | | AutoInq | [2] | not-used | | |
| | | ManualInq | [1] | | | |
| | | NoSpecifyInq | [0] | not-used | | |
| | | +0x084 | | OffsetForExpanded | [31..0] | |
| +0x088 | | - | [31..4] | reserved | | |
| | | Control | [3..0] | | | |
| +0x08C | | BitWritable | [31..0] | | | |
| +0x090 | | Value | [31..0] | | | |
| +0x094~ 0x09C | | - | - | reserved | | |
| +0x0A0 | IOLine Selector | Implemented | [31] | | | |
| | | Active | [30] | | | |
| | | - | [29..27] | reserved | | |
| | | UserSetLoadable | [26] | | | |
| | | Writable | [25] | | | |
| | | Readable | [24] | | | |
| | | ValueType | [23..16] | = 0x03 (Enumeration) | | |
| | | ControlInq | [15..0] | not-used | | |
| | | +0x0A4 | | OffsetForExpanded | [31..0] | |
| | | +0x0A8 | | Control | [31..4] | = 0x01 (Manual) |
| | | +0x0AC~ 0x0B8 | ListofElements | | [127] | VendorSpecific7 |
| | | | | | ... | ... |
| | | | | | [121] | VendorSpecific1 |
| | | | | | [120] | VendorSpecific0 |
| | [119..32] | | | not-used | | |
| | [31] | | | IOLine31 | | |
| | ... | | | ... | | |
| | [1] | | | IOLine1 | | |
| +0x0BC | Value | | [0] | IOLine0 | | |
| | | | [31..0] | | | |

Table 40 – CategoryBlock9 (DigitalIOControl) (Contd.)

| Offset | Name | Field | Bit | Description |
|------------------|------------------|-------------------|------------|----------------------|
| +0x0C0 | IOLine Source | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | |
| | | Readable | [24] | |
| | | ValueType | [23..16] | = 0x03 (Enumeration) |
| | | - | [15..5] | reserved |
| | | DefaultInq | [4] | |
| | | AutoOnceInq | [3] | not-used |
| | | AutoInq | [2] | not-used |
| | | ManualInq | [1] | |
| | | NoSpecifyInq | [0] | not-used |
| +0x0C4 | | OffsetForExpanded | [31..0] | |
| +0x0C8 | | - | [31..4] | reserved |
| | | Control | [3..0] | |
| +0x0CC~ 0x0D8 | ListOfElements | | [127..124] | not-used |
| | | | [123] | ExposureActive |
| | | | [122..116] | not-used |
| | | | [115] | FrameTransferActive |
| | | | [114..108] | not-used |
| | | | [107] | FrameActive |
| | | | [106] | FrameTriggerWait |
| | | | [105..100] | not-used |
| | | | [99] | AcquisitionActive |
| | | | [98..96] | not-used |
| | | | [95] | Timer31Active |
| | | | ... | ... |
| | | | [65] | Timer1Active |
| | | | [64] | Timer0Active |
| | | | [63..33] | not-used |
| | | | [32] | UserOutput |
| | | | [31..1] | not-used |
| | [0] | Off | | |
| +0x0DC | | Value | [31..0] | |

Table 40 – CategoryBlock9 (DigitalIOControl) (Contd.)

DigitalIOControl handles all I/O lines which the device has (including Trigger inputs). The following image is a block diagram of **DigitalIOControl**. Devices have this logic for each I/O lines.

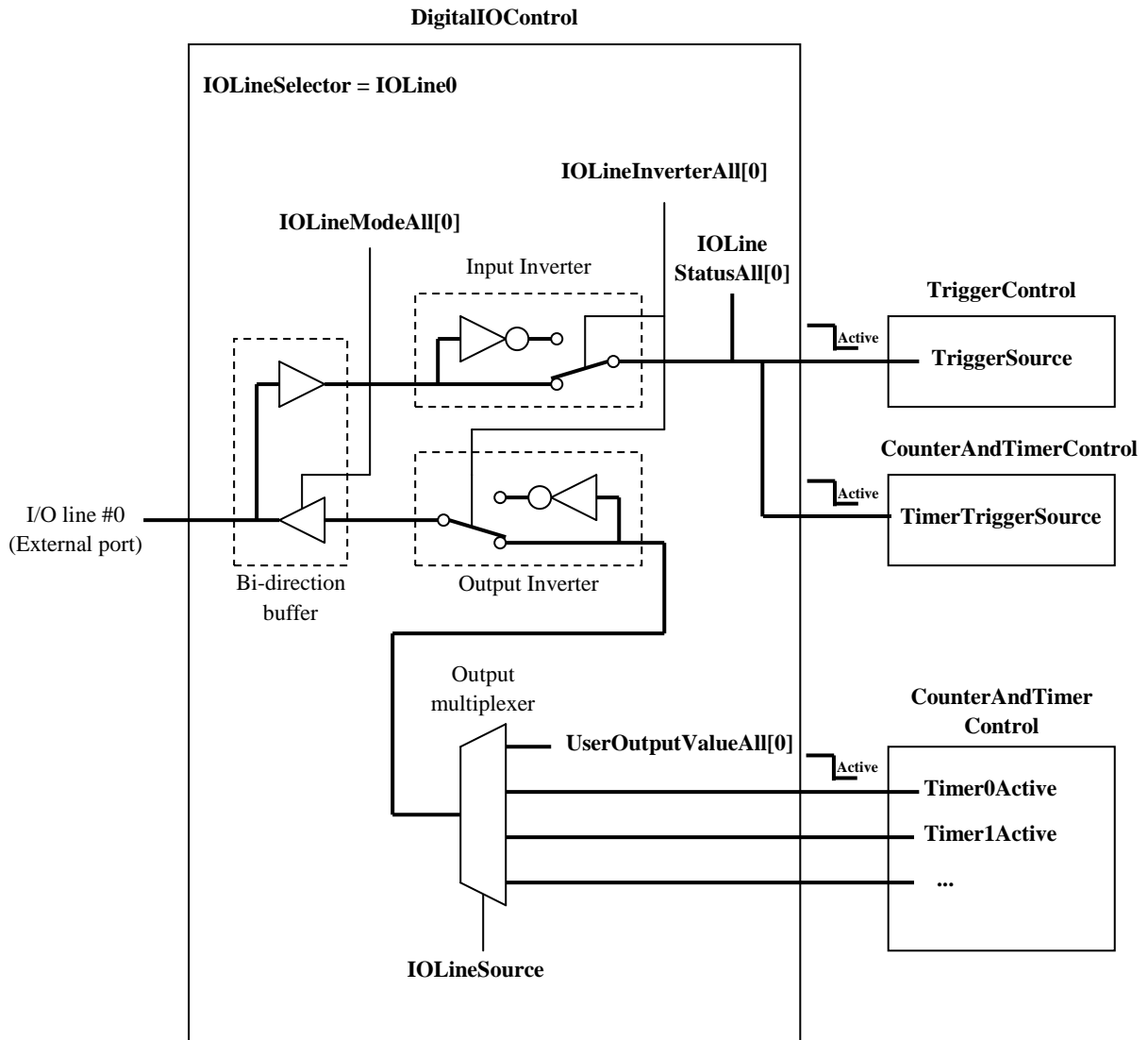


Figure 22 – Block diagram of DigitalIOControl

4.10.1 IOLineModeAll

Handles direction of I/O lines.

0 : Input

1 : Output

[R-74cd]

If device has fixed direction of I/O line, device shall turn **BitWritable** to 0 and fix Value in each corresponding bit location.

4.10.2 IOLineInverterAll

Handles the inversion of I/O lines.

0 : Not inverted

1 : Inverted

[R-75cd]

Device shall reflect it to both input and output buffers.

4.10.3 IOLineStatusAll

Indicates the current status of all I/O lines.

0 : Low

1 : High

4.10.4 UserOutputValueAll

Sets the internal register of all user outputs.

0 : Low

1 : High

4.10.5 IOLineSelector

Selects the I/O line to handle. It is reflected to **IOLineSource**.

4.10.6 IOLineSource

Selects which source signals to connect I/O line. The number of I/O line is selected by **IOLineSelector**.

Off : Output is disabled.

UserOutput : Current user output value which is handled by **UserOutputValueAll** (bit position is same as **IOLine** number).

Timer0Active, Timer1Active1,...:Current TimerValue which is handled by **FeatureCSRs** in **CounterAndTimerControl**

AcquisitionActive : Device is processing **Continuous, MultiFrame, ImageBufferRead** or **Retransmit** command of **AcquisitionCommand**.

FrameTriggerWait : Device is currently waiting a frame trigger..

FrameActive: Device is currently capturing a image.

FrameTransferActive : Device is transmitting the image data.

ExposureActive : Device is exposing

[CR-76ch]

If **IOLineSource** is **Off**, Device shall set I/O line to high-impedance.

[CR-77ch]

Host shall set bits corresponding I/O lines which are required to output in **IOLineModeAll** to **1** (Output).

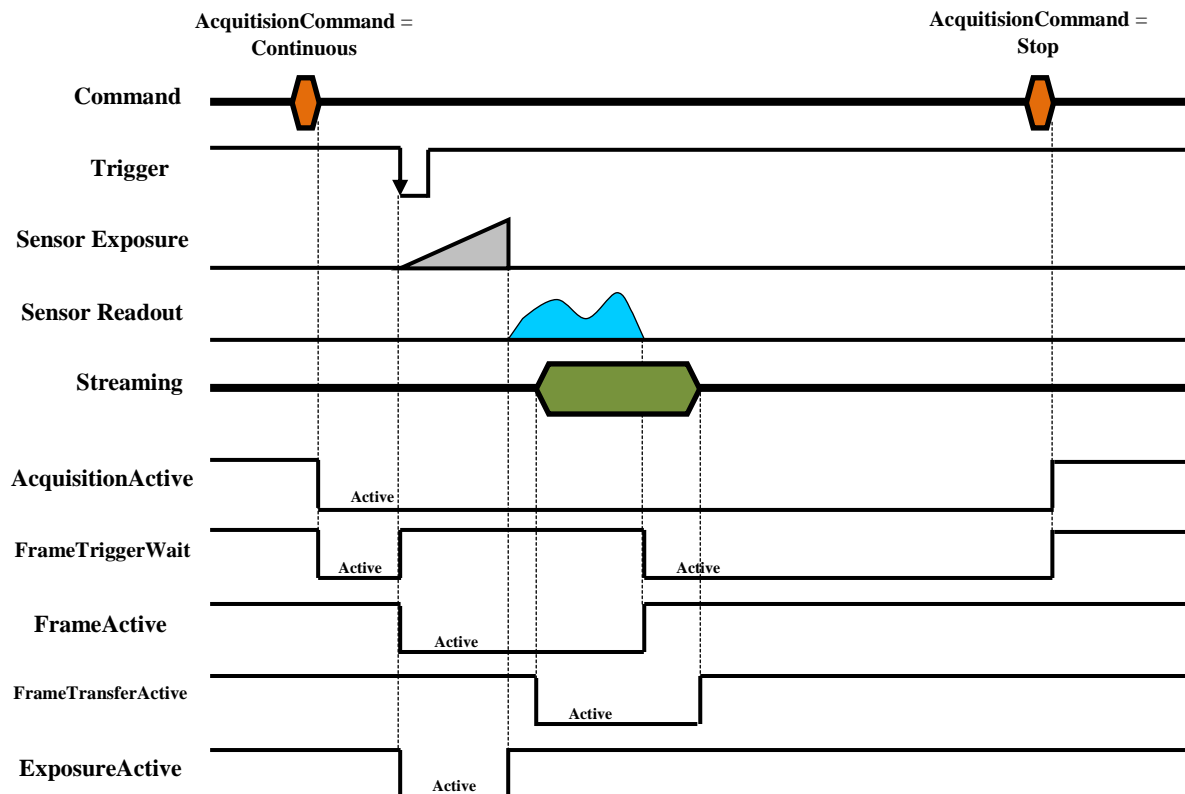


Figure 23 – Output signal definition

4.11 CategoryBlock10 (CounterAndTimerControl)

| Offset | Name | Field | Bit | Description | | |
|------------------|----------------|---------------------|-----------|----------------------|---------|-----------------|
| 0x000 | Header | CategoryBlockNumber | [31..24] | = 10 | | |
| | | SizeOfCategoryBlock | [23..0] | = 0x0000A0 | | |
| +0x004~ 0x01C | | - | - | reserved | | |
| +0x020 | Timer Selector | Implemented | [31] | | | |
| | | Active | [30] | | | |
| | | - | [29..27] | reserved | | |
| | | UserSetLoadable | [26] | | | |
| | | Writable | [25] | | | |
| | | Readable | [24] | | | |
| | | ValueType | [23..16] | = 0x03 (Enumeration) | | |
| | | ControlInq | [15..0] | not-used | | |
| | | +0x024 | | OffsetForExpanded | [31..0] | |
| | | +0x028 | | Control | [31..0] | = 0x01 (Manual) |
| +0x02C~ 0x038 | ListofElements | | [127] | VendorSpecific7 | | |
| | | ... | ... | ... | | |
| | | | [121] | VendorSpecific1 | | |
| | | | [120] | VendorSpecific0 | | |
| | | | [119..32] | not-used | | |
| | | | [31] | Timer31 | | |
| | | ... | ... | ... | | |
| | | | [1] | Timer1 | | |
| | [0] | Timer0 | | | | |
| +0x03C | | Value | [31..0] | | | |
| +0x040 | Timer Delay | Implemented | [31] | | | |
| | | Active | [30] | | | |
| | | - | [29..27] | reserved | | |
| | | UserSetLoadable | [26] | | | |
| | | Writable | [25] | | | |
| | | Readable | [24] | | | |
| | | ValueType | [23..16] | = 0x30 (Integer32) | | |
| | | - | [15..5] | reserved | | |
| | | DefaultInq | [4] | | | |
| | | AutoOnceInq | [3] | | | |
| | | AutoInq | [2] | | | |
| | | ManualInq | [1] | | | |
| | | NoSpecifyInq | [0] | | | |
| | | +0x044 | | OffsetForExpanded | [31..0] | |
| | | +0x048 | | - | [31..4] | reserved |
| +0x04C | | Control | [3..0] | | | |
| +0x050 | | Mult | [31..0] | | | |
| +0x050 | | Div | [31..0] | | | |
| +0x054 | | Min | [31..0] | | | |
| +0x058 | | Max | [31..0] | | | |
| +0x05C | | Value | [31..0] | | | |

Table 41 – CategoryBlock10 (CounterAndTimerControl)

| Offset | Name | Field | Bit | Description | |
|------------------|----------------------------|------------------------------|------------|----------------------|---------|
| +0x060~ 0x07C | Timer Duration | Same structure as TimerDelay | | | |
| +0x080 | Timer Trigger Source | Implemented | [31] | | |
| | | Active | [30] | | |
| | | - | [29..27] | reserved | |
| | | UserSetLoadable | [26] | | |
| | | Writable | [25] | | |
| | | Readable | [24] | | |
| | | ValueType | [23..16] | = 0x03 (Enumeration) | |
| | | - | [15..5] | reserved | |
| | | DefaultInq | [4] | | |
| | | AutoOnceInq | [3] | not-used | |
| | | AutoInq | [2] | not-used | |
| | | ManualInq | [1] | | |
| | | NoSpecifyInq | [0] | not-used | |
| | | +0x084 | | OffsetForExpanded | [31..0] |
| +0x088 | | - | [31..4] | reserved | |
| +0x08C~ 0x098 | | ListOfElements | Control | [3..0] | |
| | | | [127..124] | not-used | |
| | | | [125] | ExposureEnd | |
| | | | [124] | ExposureStart | |
| | | | [123..110] | not-used | |
| | | | [109] | FrameEnd | |
| | | | [108] | FrameStart | |
| | | | [107..105] | not-used | |
| | | | [104] | FrameTrigger | |
| | | | [103..102] | not-used | |
| | | | [101] | AcquisitionEnd | |
| | | | [100] | AcquisitionStart | |
| | | | [99..64] | not-used | |
| | | | [63] | IOLine31 | |
| | | | ... | ... | |
| | | | [33] | IOLine1 | |
| | | | [32] | IOLine0 | |
| [31..1] | not-used | | | | |
| [0] | Off | | | | |
| +0x09C | | Value | [31..0] | | |

Table 41 – CategoryBlock10 (CounterAndTimerControl) (Contd.)

4.11.1 TimerSelector

Selects the Timer to handle. It is reflected to **TimerDelay** and **TimerDuration**.

4.11.2 TimerDelay / TimerDuration

Handles waveform of **TimerActive** signals. Device may connect it to output multiplexer of **DigitalIOControl**. For example, it is used for strobe control. Timer is started from timer trigger event

| | | |
|-------------------------------|-----------------|--------------------------------|
| Unit of Absolute Value | | Second |
| Reference point | | - |
| Recommended value at | AutoOnce / Auto | Proper value device calculated |
| | NoSpecify | Factory setting value |

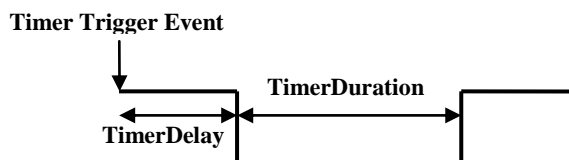


Figure 24 – Definition of TimerActive signal

4.11.3 TimerTriggerSource

Selects the timer trigger event to start timer.

Off : Timer is disabled

IOLine0~31 : Falling edge of I/O Line through **IOLineControl**.

AcquisitionStart, AcquisitionEnd : Start / stop timing of **AcquisitionActive**

FrameTrigger : Received a frame trigger.

FrameStart, FrameEnd: Start / stop timing of **FrameActive**

ExposureStart, ExposureEnd : Start / stop timing of **ExposureActive**

4.12 CategoryBlock11 (EventControl)

| Offset | Name | Field | Bit | Description |
|------------------|--------------------------------------|---------------------|--------------|------------------------|
| 0x000 | Header | CategoryBlockNumber | [31..24] | = 11 |
| | | SizeOfCategoryBlock | [23..0] | = 0x000060 |
| +0x004~ 0x01C | | - | - | reserved |
| +0x020 | Event Notification Of Frame | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | |
| | | Readable | [24] | |
| | | ValueType | [23..16] | = 0x34 (BulkBoolean32) |
| | | - | [15..5] | reserved |
| | | DefaultInq | [4] | |
| | | AutoOnceInq | [3] | not-used |
| | | AutoInq | [2] | not-used |
| | | ManualInq | [1] | |
| | | NoSpecifyInq | [0] | not-used |
| | | +0x024 | | OffsetForExpanded |
| +0x028 | | - | [31..4] | reserved |
| | | Control | [3..0] | |
| +0x02C | BitWritable | - | [31..18] | not-used |
| | | - | [17] | FrameTransferEnd |
| | | - | [16] | FrameTransferStart |
| | | - | [15..5] | not-used |
| | | - | [4] | FrameEnd |
| | | - | [3] | FrameStart |
| | | - | [2] | FrameTriggerWait |
| | | - | [1] | FrameTriggerError |
| | | [0] | FrameTrigger | |
| +0x030 | | Value | [31..0] | |
| +0x034~ 0x03C | | - | - | Reserved |

Table 42 –CategoryBlock11 (EventControl)

| Offset | Name | Field | Bit | Description |
|--------------|--------------------------------|-----------------|----------|------------------------|
| +0x040 | Event Notification Of Exposure | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | |
| | | Readable | [24] | |
| | | ValueType | [23..16] | = 0x34 (BulkBoolean32) |
| | | - | [15..5] | reserved |
| | | DefaultInq | [4] | |
| | | AutoOnceInq | [3] | not-used |
| | | AutoInq | [2] | not-used |
| | | ManualInq | [1] | |
| | | NoSpecifyInq | [0] | not-used |
| | | +0x044 | | OffsetForExpanded |
| +0x048 | | - | [31..4] | reserved |
| | | Control | [3..0] | |
| +0x04C | | BitWritable | [31..2] | not-used |
| | | | [1] | ExposureEnd |
| | | | [0] | ExposureStart |
| +0x050 | | Value | [31..0] | |
| +0x054~0x05C | | - | - | Reserved |

Table 42 –CategoryBlock11 (EventControl) (Contd.)

EventControl Handles Event feature. Host and device may be used this feature if event feature is defined in using interface (e.g. GigE-Vision, USB3 Vision or etc...).

[CR-78c]

If **EventID** is not defined by transport layer (e.g. USB3 Vision), IIDC2 device shall use as follows.

| EventID | Name |
|---------|--------------------|
| 0x8020 | FrameTrigger |
| 0x8021 | FrameTriggerError |
| 0x8022 | FrameTriggerWait |
| 0x8023 | FrameStart |
| 0x8024 | FrameEnd |
| 0x8030 | FrameTransferStart |
| 0x8031 | FrameTransferEnd |
| 0x8040 | ExposureStart |
| 0x8041 | ExposureEnd |

Table 43 – EventID list

4.12.1 EventNotificationOfFrame / EventNotificationOfExposure

Handles the event notification.

FrameTrigger : Received a frame trigger.

FrameTriggerError : a frame trigger is reached, but device cannot receive it.

FrameTriggerWait : device has ready to receive a frame trigger.

FrameStart, FrameEnd: Start / stop timing of **FrameActive**

FrameTransferStart, FrameTransferEnd: Start / stop timing of **FrameTransferActive**

ExposureStart, ExposureEnd : Start / stop timing of **ExposureActive**

4.13 CategoryBlock24~31 (VendorUniqueControl)

| Offset | Name | Field | Bit | Description |
|------------------|-----------------|---------------------|----------|-------------------|
| +0x000 | Header | CategoryBlockNumber | [31..24] | = 31 |
| | | SizeOfCategoryBlock | [23..0] | |
| +0x004~ 0x01C | | - | - | reserved |
| +0x020 | Feature CSR0 | Implemented | [31] | |
| | | Active | [30] | |
| | | - | [29..27] | reserved |
| | | UserSetLoadable | [26] | |
| | | Writable | [25] | |
| | | Readable | [24] | |
| | | ValueType | [23..16] | |
| | | - | [15..5] | reserved |
| | | DefaultInq | [4] | |
| | | AutoOnceInq | [3] | |
| | | AutoInq | [2] | |
| | | ManualInq | [1] | |
| | | NoSpecifyInq | [0] | |
| | | +0x024 | | OffsetForExpanded |
| +0x028 | | - | [31..4] | reserved |
| | | Control | [3..0] | |
| +0x02C~ 0x03C | | ValueRegisterArea | | |
| +0x040~ 0x05C | Feature CSR1 | | | |
| ... | ... | | | ... |

Table 44 – CategoryBlock24~31 (VendorUniqueControl)

VendorUniqueControl provides vendor unique control that is not related to other **BasicCSRs**. Each **FeatureCSR** has the following constraints:

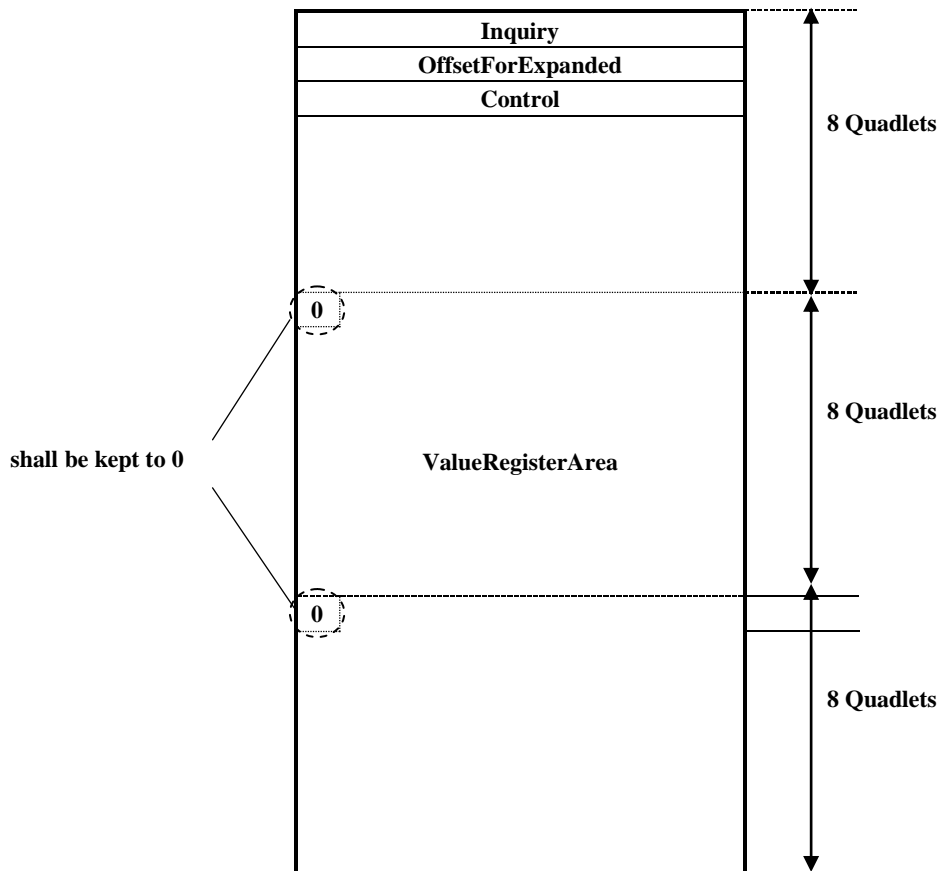


Figure 25 – FeatureCSR longer than 8 quadlets

[R-79c]

Length of **FeatureCSR** shall be in multiples of 8 quadlets.

[CR-80c]

If the actual length is less than 8 quadlets, the remaining bytes shall be reserved.

[CR-81cd]

If length is longer than 8 quadlets, the bits corresponding to **Implemented** of each **ValueRegisterArea** after 2nd 8 quadlets shall be kept to zeros.

[R-82c]

ValueType shall be **Integer32**, **PlainInteger8**, **PlainInteger32**, **PlainInteger64**, **Float32**, **Enumeration**, **BulkBoolean32**, **BulkBoolean64**, **Rectangle32**, or **String (ArrayOfPlainInteger8)**.

[R-83cd]

These **FeatureCSRs** shall not be chained from other **FeatureCSRs**. These **FeatureCSRs** may be chained to **ExpandedCSRs**

5 FeatureCSR Chaining

5.1 Use case of common logic

FeatureCSR chaining may be used for the single hardware logic which has several input ports. The following figure is an example of a device that has two **FeatureCSRs** which handle common gain amplifier hardware logic.

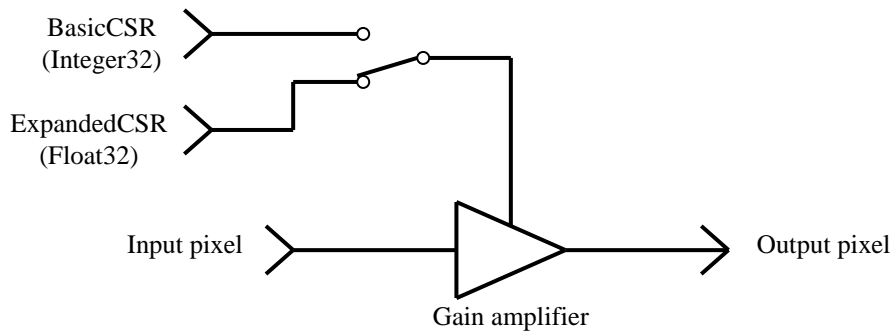


Figure 26 – Gain amplifier which has Integer32 / Float32 register

5.1.1 Value

[O-84ch]

If either one's **Value** is written, device should reflect **Value** to the other's.

[CR-85ch]

If either one's **Value** is written, device shall change **Readable** to **0** in the **FeatureCSR** which **Value** is not correct.

5.1.2 Control

[O-86ch]

If either one' is **Control** written, device shall reflect **Control** to the other's.

5.2 Use case of independent logics

FeatureCSR chaining may be used for the several hardware logics. The following figure is an example of a device that has two separate gain amplifier hardware logics.

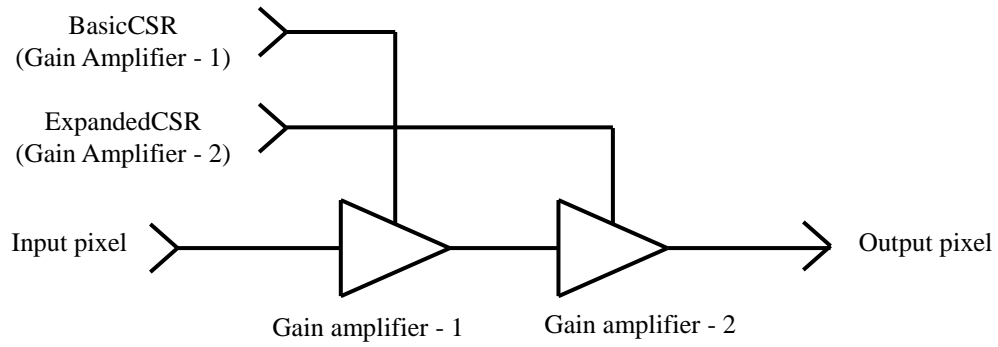


Figure 27 –2 gain amplifier logics

5.2.1 Value

These are independent.

5.2.2 Control

These are independent.

Annex A

Recommended FeatureCSRs of camera devices

The following table describes the recommended FeatureCSRs of minimum capability camera.

| FeatureCSR | Description |
|-------------------------------------|--|
| CategoryBlock2 (ImageFormatControl) | |
| ImageFormatSelector | Camera should have at least one image format. |
| ApplyImageFormat | Camera should support Done status. |
| ImageSize | Camera should have at least one pattern of image size. |
| PixelCoding / PixelSize | Camera should support at least one pair of PixelCoding and PixelSize. |
| CategoryBlock3 (AcquisitionControl) | |
| Acquisition Command | Camera should support either Stop or Abort command, and support any one of Continuous, MultiFrame or ImageBufferRead at least. |

Table 45 – Recommended FeatureCSRs

Annex B**Supporting multiple-ROI**

IIDC2 supports multiple-ROI handling. This section describes recommended implementing method.

B.1 Overview

An example of multiple-ROI is as following figure.

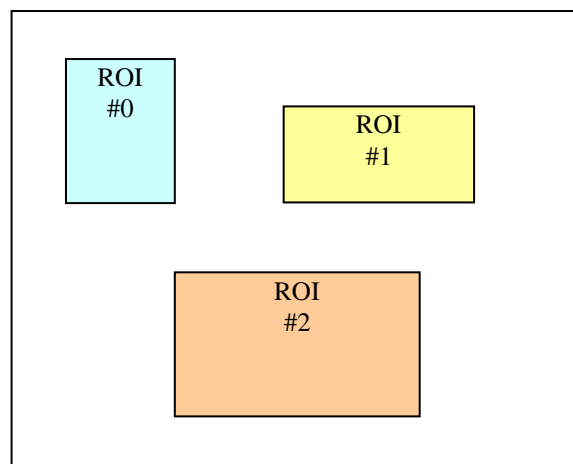


Figure 28 – Example of multiple-ROI

The condition of multiple-ROI is as follows.

- All regions are shaped rectangle
- There is flexibility of using number of ROIs.
- Streaming data format is defined on transport layer (there is no information in this section)

Multiple-ROI is implemented as **ExpandedCSR** of **ImageSize**. The block diagram is as following figure.

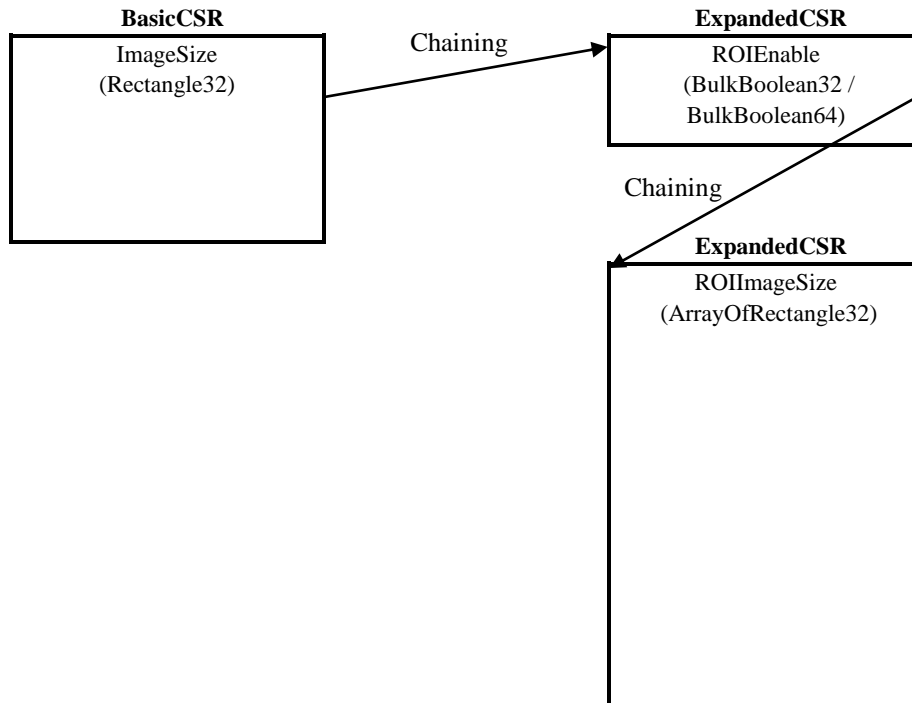


Figure 29 – Structure of multiple-ROI

B.2 BasicCSR

[CO-87cd]

For backward compatibility, device SHOULD support **BasicCSR** handling (1 ROI mode).

B.3 ROIEnable

Handles which ROIs are used. **1** is enable, **0** is disable. **ValueType** is **BulkBoolean32** (up to 32 ROIs) or **BulkBoolean64** (up to 64 ROIs). A bit position indicates ROI number.

B.4 ROIImageSize

Sets image size and position of each ROIs. An index of array indicates ROI number.

Annex C
Keeping compatibility for previous version of IIDC2

For backward compatibility, **IIDC2** device / host should support previous specification.

C.1 CSR Chaining mechanism (for host)

IIDC2 Ver.1.0.0 has a little bit different structure from the section **3.2(General structure)**. It is described below.

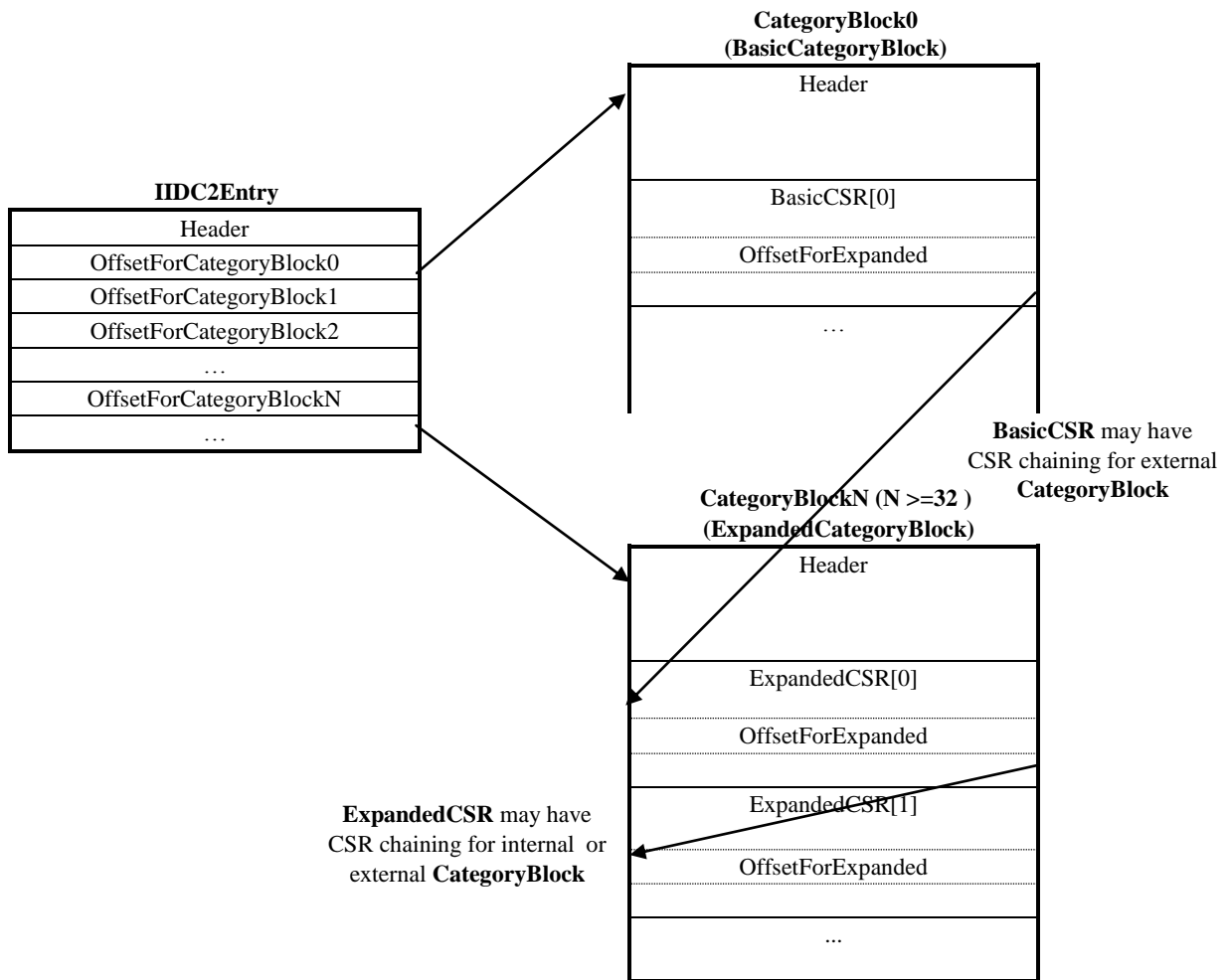


Figure 30 – CSR Chaining of IIDC2 Ver.1.0.0

IIDC2 Ver.1.0.0 has **ExpandedCategoryBlocks** at upper of **CategoryBlock32**. All **ExpandedCSRs** are placed in **ExpandedCategoryBlocks**.

And **OffsetForExpanded** has **CategoryBlockNumber** field which informs chaining destination of **CategoryBlock**.

| Bit | Field | Description |
|----------|---------------------|--|
| [31..24] | CategoryBlockNumber | Indicates the index of CategoryBlock which exist in chained ExpandedCSR . If this field is zero, chaining ExpandedCSR is in the same CategoryBlock . |
| [23..0] | Offset | Indicates the offset quadlets of chaining ExpandedCSR from the top of CategoryBlock . |

Table 46 – OffsetForExpanded in IIDC2 Ver.1.0.0

[CO-88ch]

If compatibility for IIDC2 Ver.1.0.0 is required, host should support CSR chaining mechanism using **CategoryBlockNumber** field.

Document history

| Version | Section / Part | Description |
|---|--|--|
| 1.0.0 | All | First version |
| 1.1.0 | All | Added Requirements Terminology (Eliminated Conformance) |
| | 2.1.2 Glossary, 3.5.4.1 Integer | Added AbsoluteValue and description |
| | 3.2 General structure, 3.5.2 OffsetForExpanded | Changed CSR chaining mechanism, eliminated description of ExpandedCategoryBlock |
| | 3.3 IIDC2Entry (offset list of CategoryBlock), 3.4.1 CategoryBlock List, 4 Device Control Register | Added CategoryBlock11, CategoryBlock24~30 |
| | 3.3.4 OffsetForXmlManifestTable, 3.3.5 OffsetForCategoryBlock0 ~ 31, 3.5.2 OffsetForExpanded | Added information of offset address, changed unit of address offset (Quadlets to Bytes) |
| | 3.4.2 Header of CategoryBlock, 4 Device Control Register | Changed unit of SizeOfCategoryBlock (Quadlets to Bytes) |
| | 3.5.3 Control, 3.5.4.4 Enumeration, 4 Device Control Register | Included Technical Bulletin JIIA CPR-003-2013 / TB2013001 |
| | 4 Device Control Register | Added information of setting Implement to 0 ([CR-43ch]) |
| | 4.1 CategoryBlock0 (DeviceControl) | Added new feature of LEDIndicatorLuminance |
| | 4.1.1 DeviceReset, 4.1.2 DevicePower, 4.3.1 ImageFormatSelector, 4.7.1 LUTEnable, 4.7.2 LUTBankSelector, 4.7.3 LUTValueAll, 4.8.1 TriggerMode, 4.8.2 TriggerSequence, 4.8.3 TriggerSource, 4.8.6 SoftwareTrigger, 4.9.2 UserSetSelector / UserSetCommand, 4.10.5 IOLineSelector, 4.11.1 TimerSelector | Added elements of VendorSpecific0~7 |
| | 4.3 CategoryBlock2 (ImageFormatControl) | Added new features of BinningHorizontal / BinningVertical, DecimationHorizontal / DecimationVertical, ReverseX / ReverseY and TestPattern |
| | 4.3.4 PixelCoding / PixelSize | Added elements of Mono10 / 12, BayerGR10 / 12, BayerRG10 / 12, BayerGB10 / 12, BayerBG10, 12 and BGRPacked Eliminated elements of Mono16, MonoSigned16, RGB16, RGBSigned16, BayerGR16, BayerRG16, BayerGB16 and BayerBG16 |
| | 4.8.2 TriggerSequence | Added element of Sequence6 (Frame Burst mode) |
| | 4.9 CategoryBlock8 (UserSetControl) | Added UserSetDefault |
| | 4.10.6 IOLineSource | Added elements of AcquisitionActive, FrameTriggerWait, FrameActive, FrameTransferActive, ExposureActive and Figure 23 – Output signal definition |
| | 4.11 CategoryBlock10 (CounterAndTimerControl) | Added new feature of TimerTriggerSource |
| | CategoryBlock11 (EventControl) | Added new CategoryBlock and FeatureCSRs |
| 5.2.2 Annex C Keeping compatibility for previous version of IIDC2 | Added section | |

| | | |
|--|-----------------------------|----------------------------|
| | Transport layer of IEEE1394 | Eliminated all information |
|--|-----------------------------|----------------------------|