IDEAS Readout and Control Packet Protocol Reference

Table of Contents

1 Document Description

1.1 Document Scope

This document describes the IDEAS Readout and Control Protocol and serves as a reference for implementation.

1.2 Document Change Log

Table 1: Document change log.

Date	Version	Responsible	Comment
2016-03-02	V1R0 (Draft)	D. Steenari, IDEAS	Reissue of old document, with new structure. Old versioning / document is deprecated.
2016-04-29	V1R1 (Draft)	D. Steenari, IDEAS	Added additional packet formats.
2016-05-23	V1R2 (Draft)	D. Steenari, IDEAS	Added ASIC SPI Register packets.
2016-10-03	V1R3 (Draft)	D. Steenari, IDEAS	Added Multi-Event Pulse-Height data packet. Added separate section for trigger types (to be revised). Numerous minor updates.
2017-04-20	V1.6	D. Steenari, IDEAS	Connecting version of reference document with IDEAS Packet Library release version.
2017-05-10	V1.7	Jan-Erik Holter, IDEAS	Updated the Packet Type 0xD1, Image Data.
2022-10-24	V1.8	A.E.Olsen, IDEAS	Removed deprecated 0xD9 data packet, added new 0xDA GDS100 specific data pack.
2023-01-16	V1.6	A.E.Olsen, IDEAS	Corrected imaging and pipeline packet layout descriptions.

1.3 List of Acronyms

ASIC	Application Specific Integrated Circuit
DAC	Digital-to-analogue converter
RO	Read-out
TC	Telecommand
TCP	Transmission Control Protocol
TBC	To be confirmed
TBD	To be determined
TM	Telemetry
UDP	User Datagram Protocol

2 Readout and Control Protocol

For each detector / readout system two network sockets are used:

- TCP for control and monitoring – the detector is the network server (read-out PC is the client).
- UDP for read-out frame data – read-out PC is the network server (the detector / readout system is the client).

2.1 Packet Encapsulation – Ethernet Network

The MTU (Maximum Transmission Unit) for Ethernet is 1500 bytes.

2.1.1 Packet Encapsulation – TCP

Each TCP packet shall contain exactly one data packet.

2.1.2 Packet Encapsulation – UDP

Each UDP datagram shall contain exactly one data packet.

2.2 Packet Format

The packet format is the same for both TCP and UDP communication.

Table 2: Packet Format.

Packet Header	Packet Data
Mandatory header Format: fix fields	Optional data Format: Depending on "Packet Type" in the Packet Header
10 bytes	0 to 1500 bytes

Where the fields are as follows:

- Packet Header – Fix length. Dependent on the source of the packet. Defined in section 2.2.1.
- Packet Data – Variable length. Dependent on the "Packet Type" field in the Packet Header. The maximum packet length is dependent on the medium the packets are transferred over.

2.2.1 Packet Header Format

The Packet Header format is a fixed length number of parameters. Which are different depending on the source of the packet.

Table 3: Packet Header – System to PC packets.

Packet Header						
Packet ID			Packet Sequence		Timestamp	Data Length
Version	System Number	Packet Type	Sequence Flag	Packet Count		
Fixed	Enum	Enum	Enum	Uint14	Uint32	Uint16
3 bits	5 bits	8 bits	2 bits	14 bits	32 bits	16 bits
2 bytes			2 bytes		4 bytes	2 bytes
10 bytes						

Table 4: Packet Header – PC to system packets.

Packet Header						
Packet ID			**Packet Sequence**		**Reserved**	**Data Length**
Version	**System Number**	**Packet Type**	**Sequence Flag**	**Packet Count**		
Fixed	Enum	Enum	Enum	Uint14	Fixed = 0	Uint16
3 bits	5 bits	8 bits	2 bits	14 bits	32 bits	16 bits
2 bytes			2 bytes		4 bytes	2 bytes
10 bytes						

Packet Header field descriptions:

- Packet ID field – Enumerated 16 bits, with sub-fields below.
  - Version – Version of packet protocol. 3 bits fixed value = 0b000.
  - System Number – unique ID number of system.
  - Packet type – specifies how the "Packet data" field should be decoded (see sections below).

- Packet sequence – specifies what sequence (e.g. for read-out data, there will be a series of packets all in the same sequence) the packet is a part of:
    - 0b00 – *StandAlone* – packet is the only packet in the sequence.
    - 0b01 – *FirstPacket* – packet is the first packet in a sequence.
    - 0b10 – *ContinuationPacket* – packet continues a started sequence.
    - 0b11 – *LastPacket* – packet is the last packet in the sequence.

- Packet counter – Incremental counter for packet identification (one counter per system). Values: 0 to 0x3FFF
- Time stamp – System timestamp when packet was created. Values 0-2^{32}.
- Data length – Defines the length of the "Packet Data" field in number of bytes.

2.2.2 Packet Types

The “Packet Type” field of the Packet Header defines how the Packet Data field shall be decoded.

Table 5: List of packet types.

Packet type	TC / TM	Target	Name	Description
0x10	TC	System	Write System Register	Writes to a system register
0x11	TC	System	Read System Register	Reads a system register
0x12	TM	PC	System Register Read-Back	Read-back packet to be sent after a “Write System Register” or “Read System Register”
0xC0	TC	ASIC	ASIC Configuration Register Write/Read	Write/Read to an ASIC configuration register
0xC1	TM	PC	ASIC Configuration Register Read-Back	Read-back packet to be sent after a “ASIC configuration register write/read”
0xC2	TC	System	ASIC SPI Register Write	Write an ASIC internal SPI register.
0xC3	TC	System	ASIC SPI Register Read	Read an ASIC internal SPI register.
0xC4	TM	PC	ASIC SPI Register Read-Back	Read-back packet send after an “ASIC SPI Register Write” or an “ASIC SPI Register Read”.
0xD0	TM	PC	Counting Frame Data	Frame read-out data for counting systems. Applies to CA-type ASICs.
0xD1	TM	PC	Image Data	Image data format from camera products.
0xD4	TM	PC	Multi-Event Pulse-Height Data Packet	Single-Event read-out data packet for pulse-height (spectroscopic) systems. Applies to IDE3- and IDE4-type ASICs.
0xD5	TM	PC	Single-Event Pulse-Height Data Packet	Single-Event read-out data packet for pulse-height (spectroscopic) systems. Applies to IDE3- and IDE4-type ASICs.
0xD6	TM	PC	Trigger Time Data Packet	Event time of trigger per channel. Multi-event packet.
0xDA	TM	PC	Extended GDS Data Packet	Variable data length packet. Header contains identification of the event and channel. The data field contains samples from one channel.

2.3 Packet Data Fields: Control and Read-Back Packets.

2.3.1 0x10 — Write System Register Packet

The “Write System Register” packet type is used by the PC to configure internal register settings in the system. The system will respond with a “System Register Read-Back” packet after the register write has been performed.

Table 6: Write System Register packet fields.

Register Address	Register Length	Register Data
2 bytes	1 byte	1 - 256 bytes
4 – 259 bytes		

Write System Register packet field descriptions:

- Register Address – Register map address for the register that is to be written.
- Register Length – Length of the data (in bytes) to write.
- Register Data – Variable length (set by the Register Length field) of register data to write.

2.3.2 0x11 — Read System Register Packet

The “Read System Register” packet type is used by the PC to monitor internal register settings in the system. The system will respond with a “System Register Read-Back” packet after the register read has been performed.

Table 7: Read System Register packet fields.

Register Address
2 bytes
2 bytes

Read System Register packet field descriptions:

- Register Address – Register map address for the register that is to be read.

2.3.3 0x12 — System Register Read-Back Packet

The “System Register Read-Back” packet type is used by the system as a response packet after a “Write System Register” or a “Read System Register” packet has been received from the PC. The packet contains the current value of a system register.

Table 8: System Register Read-back packet fields.

Register Address	Register Length	Register Data
2 bytes	1 byte	1 - 256 bytes
4 – 259 bytes		

System Register Read-Back packet field descriptions:

- Register Address – Register map address for the register that was read.
- Register Length – Length of the data (in bytes) that was read.
- Register Data – Variable length (set by the Register Length field) of read-back register data.

2.3.4 0xC0 — ASIC Configuration Register Write/Read Packet

The "ASIC Configuration Register Write/Read" packet type is used to set the configuration on a single or a multiple ASICs (depending on the hardware configuration of the system, e.g., ASIC daisy changing).

Table 9: ASIC Configuration Register Write/Read packet fields.

ASIC(s) ID	Configuration Length	Configuration Data
1 byte	2 bytes	1 - 1024 bytes
4 – 1027 bytes		

ASIC Configuration Register Write/Read packet field descriptions:

- ASIC(s) ID – ID number of ASIC(s) to configure.
- Configuration Length – Length (in bits) of the configuration to send.
- Configuration Data – ASIC configuration data to send to the selected ASIC(s). Number of bits of configuration data shall be according to the Configuration Length field, padded with 0s to the nearest byte multiple.

2.3.5 0xC1 — ASIC Configuration Register Read-Back Packet

The "ASIC Configuration Register Read-Back" packet type is sent by the system to the PC as a response to an "ASIC Configuration Register Write/Read" packet.

Table 10: ASIC Configuration Register Read-Back packet fields.

ASIC(s) ID	Configuration Length	Configuration Data
1 byte	2 bytes	1 - 1024 bytes
4 – 1027 bytes		

ASIC Configuration Register Write/Read packet field descriptions:

- ASIC(s) ID – ID number of ASIC(s) that was configured.
- Configuration Length – Length (in bits) of the configuration sent.

- Configuration Data – ASIC configuration that was sent to the selected ASIC(s). Number of bits of configuration data shall be according to the Configuration Length field, padded with 0s to the nearest byte multiple.

2.3.6 0xC2 — ASIC SPI Register Write Packet

The "ASIC SPI Register Write Packet" packet is used to set the configuration in a single SPI register in an ASIC or multiple ASICs.

Table 11: ASIC SPI Register Write packet fields.

ASIC(s) ID	SPI Format	ASIC Register Address	ASIC Register Bit Length	ASIC Register Data
1 byte	1 byte	2 bytes	2 bytes	1 - 1024 bytes
7 – 1030 bytes				

ASIC SPI Register Write packet field descriptions:

- ASIC(s) ID – ID number of ASIC(s) to configure.
- SPI Format – SPI format to use for the transaction.
  - ASIC SPI Format = 0x01: IDE3380 SIPHRA SPI Format
- ASIC Register Address – Internal ASIC address of register to write.
- ASIC Register Bit Length – Length (in bits) of the register to write.
- ASIC Register Data – Data to write to ASIC register. Number of bits of configuration data shall be according to the ASIC Register Bit Length field, padded with 0s to the nearest byte multiple.

2.3.7 0xC3 — ASIC SPI Register Read Packet

The "ASIC SPI Register Read Packet" packet is used to read the configuration in a single SPI register in an ASIC or multiple ASICs.

Table 12: ASIC SPI Register Read packet fields.

ASIC(s) ID	SPI Format	ASIC Register Address	ASIC Register Bit Length
1 byte	1 byte	2 bytes	2 bytes
6 bytes			

ASIC SPI Register Write packet field descriptions:

- ASIC(s) ID – ID number of ASIC(s) to read register from.
- SPI Format – SPI format to use for the transaction (See Section 2.3.6).

- ASIC Register Address – Internal ASIC address of register to read.
- ASIC Register Bit Length – Length (in bits) of the register to read.

2.3.8 0xC4 — ASIC SPI Register Read-Back Packet

The "ASIC SPI Register Read-Back Packet" packet is is sent by the system to the PC as a response to a "ASIC SPI Register Write Packet" or an "ASIC SPI Register Read Packet".

Table 13: ASIC SPI Register Write packet fields.

ASIC(s) ID	SPI Format	ASIC Register Address	ASIC Register Bit Length	ASIC Register Data
1 byte	1 byte	2 bytes	2 bytes	1 - 1024 bytes
7 – 1030 bytes				

ASIC SPI Register Write packet field descriptions:

- ASIC(s) ID – ID number of ASIC(s) that was configured.
- SPI Format – SPI format to use for the transaction (See Section 2.3.6).
- ASIC Register Address – Internal ASIC address of register that was read.
- ASIC Register Bit Length – Length (in bits) of the register that was read.
- ASIC Register Data – Data that was read from the ASIC register. Number of bits of configuration data is according to the ASIC Register Bit Length field, padded with 0s to the nearest byte multiple.

2.4 Packet Data Fields: Data Packets

2.4.1 0xD0 — Counting Frame Data Packet (Obsolete)

The "Counting Frame Data" packet type contains the read-out data from the event counters and the overflow bits. Data from 7 ASICs are sent per packet.

Table 14: Counting Frame Data packet fields.

Data ID	Cnt. Grp. #1	OF Grp. #1	Cnt. Grp. #2	OF Grp. #2	Cnt. Grp. #3	OF Grp. #3	Cnt. Grp. #4	OF Grp. #4	Cnt. Grp. #5	OF Grp. #5	Cnt. Grp. #6	OF Grp. #6	Cnt. Grp. #7	OF Grp. #7
1 byte	128 bytes	8 bytes	128 bytes	8 bytes	128 bytes	8 bytes	128 bytes	8 bytes	128 bytes	8 bytes	128 bytes	8 bytes	128 bytes	8 bytes
953 bytes														

Read-out Frame Data packet field descriptions:

- Data Group – ID number of which group the data in the packet belongs. Two possible values:
  - '0' = Data Group 1
  - '1' = Data Group 2

- Counter Group (Cnt. Grp.) #*N* – Group of 64 counters, each of 16 bits, all belonging to channels of the same ASIC.
    - The source of the counter data depends on the value of the Data ID field, see Table 15 below.
    - The order of the read-out counters is according to the order of the ASIC channels (0-63 per ASIC).

- Overflow Flags Group (OF. Grp.) #*N* – Group of 64 overflow flags (OF), each of 1 bit, all belonging to channels of the same ASIC.

Table 15: ASIC data order (Group #) vs. the Data ID field in Counting Frame Data packet fields.

Counting Frame Data Packet Field	Data ID = 0	Data ID = 1
Grp. #1	ASIC-1A	ASIC-7B
Grp. #2	ASIC-3A	ASIC-2A
Grp. #3	ASIC-5A	ASIC-4A
Grp. #4	ASIC-7A	ASIC-6A
Grp. #5	ASIC-1B	ASIC-2B
Grp. #6	ASIC-3B	ASIC-4B
Grp. #7	ASIC-5B	ASIC-6B

2.4.2 0xD1 — Image Data Packet

The “Image Data” packet is a general-purpose data format for hyper-spectral image data. Image data can be split over multiple packets.

Table 16: Image Data Packet fields.

Frame Number	Image Width (pixels)	Image Height (pixels)	Number of Spectral Channels	Reserved	Data Width (bits)	User Defined	Number of Data Packets per Image	Current Data Packet Sequence	Reserved	Image Data
2 bytes	2 bytes	2 bytes	2 bytes	1 Byte	1 byte	4 bytes	2 bytes	2 bytes	2 bytes	0 – 1400 bytes
20 – 1420 bytes										

Image Data packet field descriptions:

- Frame Number – The Frame number of the image.
- Image Width (pixels) – The full image width (x-axis) in number of pixels.
- Image Height (pixels) – The full image height (y-axis) in number of pixels.
- Number of Spectral Channels – Number of spectral channel frames per image.

- Data Width (bits) – The data width for each sample, e.g., 8-bit, 12-bit, 16-bit, etc.
- User Defined – Please check the product documentation for a description of this field.
- Number of Data Packets per Image – The total number of data packets per full image (all spectral channels).
- Current Data Packet Sequence – The data sequence the current data belongs to.
- Image Data – Sequential image data. The position of the current image data is specified by the meta data fields above. The length of the data is given by the packet data length field (in the packet header).

Example:

- A multi-spectral 16-bits image with size 112 x 4 pixels and 6 spectral channels is transferred. Total number of pixels is 448. Total number of data points is 2688 (448x6). Total data size is 5376 bytes. A total of 4 packets is needed to transfer the data (5376 / 1400 = 3.84).
  Fields values:
  - Image width (pixels) = 112
  - Image height (pixels) = 4
  - Number of Spectral Channels = 6
  - Data Width = 16
  - Number of Data Packets per Image = 4
  - Current Data Packet Sequence = [0;1;2;3]
  - Image data = [1400 bytes array; 1400 bytes array; 1400 bytes array; 1176 bytes array]

2.4.3 0xD4 — Multi-Event Pulse-Height Data Packet

The “Multi-Event Pulse-Height Data” packet is sent from the system after several independent readouts of an (or multiple) ASICs – i.e., the events are collated to a single packet to reduce overhead

Table 17: Multi-Event Pulse-Height Data Packet fields.

<table>
<tr><td colspan="2">Packet Data Field Header</td><td colspan="7">Event #1</td><td rowspan="4"></td><td rowspan="3">Event #N</td></tr>
<tr><td rowspan="2">Number of Events (N)</td><td rowspan="2">Number of Samples / Event (M)</td><td>Event Header</td><td colspan="4">Event Sample Data #1</td><td rowspan="2">…</td><td rowspan="2">Event Sample Data #M</td></tr>
<tr><td>Timestamp</td><td>Trigger Type</td><td>Source ID</td><td>Channel ID</td><td>Sample</td></tr>
<tr><td rowspan="2">1 byte</td><td rowspan="2">2 bytes</td><td rowspan="2">4 bytes</td><td>1 byte</td><td>1 byte</td><td>1 byte</td><td>2 bytes</td><td rowspan="2">…</td><td rowspan="2">5 bytes</td></tr>
<tr><td colspan="4">5 bytes</td></tr>
<tr><td colspan="2" rowspan="2">3 bytes</td><td colspan="7">4 + 5*M bytes</td><td>4 + 5*M bytes</td></tr>
<tr><td colspan="8">(4 + 5*M) * N bytes</td></tr>
<tr><td colspan="10">12 – 1450 bytes</td></tr>
</table>

Multi-Event Pulse-Height Data packet field descriptions:

- Source ID – ID of ASIC or chain of ASIC that was readout.
- Trigger Type – Contains the source that caused the trigger / event (and the readout). See Section 2.5.
- Channel ID – ID of (last) readout channel.
- Number of Samples – Number of 16-bit samples in the “Samples” field. 1 to 741 samples.
- Samples – Readout sample data array, each 16-bit. Up to 741 samples.

2.4.4 0xD5 — Single-Event Pulse-Height Data Packet

The “Single-Event Pulse-Height Data” packet is sent from the system at each readout of an (or multiple) ASICs. The trigger-source that generated the event could be the ASIC internal triggering (due to a physical event in the detector), a forced readout or through a calibration readout.

Table 18: Single-Event Pulse-Height Data Packet fields.

Source ID	Trigger Type	Channel ID	Hold Delay	Number of Samples	Samples
1 byte	1 byte	1 byte	2 bytes	2 bytes	2 – 1482 bytes
9 – 1489 bytes					

Pulse-Height Data packet field descriptions:

- Source ID – ID of ASIC or chain of ASIC that was readout.
- Trigger Type – Contains the source that caused the trigger / event (and the readout). See Section 2.5.
- Channel ID – ID of (last) readout channel.
- Hold delay – Used hold delay at time of event.
- Number of Samples – Number of 16-bit samples in the “Samples” field. 1 to 741 samples.
- Samples – Readout sample data array, each 16-bit. Up to 741 samples.

2.4.5 0xD6 — Trigger Time Data Packet

The “Trigger Time Data” packet is sent from the system at each readout of an (or multiple) ASICs. The trigger-sources that generates the event are trigger outputs from the ASIC.

Table 19: Trigger Time Data Packet fields.

Number of Events	Event 1 Timestamp	Event 1 Triggered ASIC + Channel	…	Event N Timestamp	Event N Triggered ASIC + Channel
1 byte	4 bytes	1 byte (2+6 bits)		4 bytes	1 byte (2+6 bits)

1 + 5*N bytes

Trigger Time Data packet field descriptions:

- Number of Events – Number of events in packet (N) – 1.
- Triggered Channel – Contains the trigger number.

2.4.6 0xDA — Pipeline Sampling Data Packet

The packet is sent from the system at each readout of a single channel in a pipeline sampling ASIC. A full readout event of multiple channels is built up from a sequence of Pipeline Sampling Data packets, using the packet sequence fields in the packet header.

Table 20: Pipeline Sampling Data Packet fields.

<table>
<tr><th rowspan="2">Source ID</th><th rowspan="2">Trigger Type</th><th rowspan="2">Status</th><th colspan="2">ASIC Dout</th><th rowspan="2">Event ID</th><th rowspan="2">PPS Timestamp</th><th colspan="3">ASIC Channel Data</th></tr>
<tr><td>Header</td><td>Data</td><td>Cell 0 Data</td><td>…</td><td>Cell 159 Data</td></tr>
<tr><td rowspan="3">1 byte</td><td rowspan="3">1 byte</td><td rowspan="3">2 bytes</td><td>5 bits</td><td>11 bits</td><td rowspan="3">4 bytes</td><td rowspan="3">4 bytes</td><td>2 bytes</td><td>…</td><td>2 bytes</td></tr>
<tr><td colspan="2">2 bytes</td><td colspan="3">320 bytes</td></tr>
<tr><td colspan="10">334 bytes</td></tr>
</table>

The fields in the packet data field:

- Source ID – ID of ASIC or chain of ASIC that was readout.
- Trigger Type – Contains the source that caused the trigger / event (and the readout). See Section 2.5.
- Status – Chosen by user. For instance, hold delay or number of transmitted channels.
- ASIC Dout – ID of channel data originated from. Corresponds to Dout from ASIC, (with sub-fields specified below).
- Event ID – Data field set by Trigger interface.
- PPS timestamp – Number of Clock cycles from last periodic pulse to trigger (Periodic Pulse Stamp).
- Channel data – Digitized channel data. 160 values, corresponding to the sample and hold pipeline cells of the channel.
    - Most significant bit = ADC overflow
    - Least significant 14 bits = ADC value

ASIC Dout field (determined from Dout header), decode according to ASIC data sheet.

- Cathode channel:

Table 21: Pipeline Sampling Data Packet, ASIC Dout field for cathode channel.

Dout header	Reserved	Cell pointer
Fixed value = 0b01100	Fixed value = 0b100	uint8
5 bits	3 bits	8 bits
2 bytes		

  - Dout header – 4 bits, fixed value = **0b1100** for cathode channel.
  - Cell pointer – Shows the column pipeline pointer at the time of the trigger. Values: 0-159

Table 22: Pipeline Sampling Data Packet, ASIC Dout field for anode channel.

Dout header	Trigger flag	(Padding)	X address	(Padding)	Y address
Fixed value = 0b01101	Enum	Fix val. = 0b0	uint4	Fix val. = 0b0	uint4
5 bits	1 bit	1 bit	4 bits	1 bit	4 bits
2 bytes					

  - Dout header – 4 bits, fixed value = **0b1101** for anode channels.
  - Trigger flag – If ASIC is not in “full readout” mode: specifies if the channel triggered the event. 0b1 if triggered, 0b0 if not triggered. If ASIC is in “full readout” mode: always equal to 0b1.
  - X address – 4-bit unsigned integer, specifying the x-address of the channel.
  - Y address – 4-bit unsigned integer, specifying the y-address of the channel.

Event ID 32-bit word from GSC- trigger module

2.5 Trigger Types

The "Trigger Type" field, used in multiple data packets, defines the origin of the event that caused the readout to happen.

The field can take on of the following values:

- Trigger Type = 0x00 – System auto forced readout (triggered by system clock)
- Trigger Type = 0x01 – Software trigger (by user)
- Trigger Type = 0x02 – ASIC trigger (caused by physical event)
- Trigger Type = 0x03 – Calibration trigger
- Trigger Type = 0x04 – External trigger
- Trigger Type = 0x05 – Single channel trigger mode
- Trigger Type = 0x06 – XA Multi trigger

3 System Register Map

The system register map contains a list of the registers that can be read, written, or pulsed using the "Write System Register" and "Read System Register" packet types.

3.1 Mandatory System Registers

Below is a list of system registers that are always present in IDEAS products.

Table 23: System registers that are always present.

Group	Name	Address	Length (bits)	Type	Def value	Description
System	SerialNumber	0x0000	32	R	*N/A*	Unique serial number of unit.
System	FirmwareType	0x0001	16	R	*N/A*	Type of firmware, i.e., which ASIC/system the firmware is for.
System	FirmwareVersion	0x0002	16	R	*N/A*	Firmware version number.
System	SystemNumber	0x0010	5	R/W	*N/A*	System ID, 5-bit value. Sent in packet header field.
System	ReadoutPacketCounter	0xF008	14	R/W	0	The current value of the readout packet counter (i.e., next data packet will have this value in the "Packet Count" field.

3.2 Common Specific System Registers

Below is a list of common system registers that are often present in IDEAS products.

Table 24: Example system register map that are often present in IDEAS products.

Group	Name	Address	Length (bits)	Type	Def value	Description

3.3 Product Specific System Registers

Below is a list of example system registers that are only present in specific IDEAS products.

Table 25: Example system registers present in some specific IDEAS products.

Group	Name	Address	Length (bits)	Type	Def value	Description

4 Appendix A – Example Packets

To be added.

4.1 ASIC Configuration Packet Example: Daisy chained ASICs

Table 26: Example ASIC configuration packet data field, for two daisy chained ASICs: ASIC-1A and ASIC-1B. Where 1A is the first ASIC in the chain (i.e. the ASIC with the reg_in port connected to the FPGA/Controller).

<table>
<tr><th>ASIC(s) ID field</th><th>Configuration Length</th><th colspan="9">Configuration Data</th></tr>
<tr><td rowspan="2">0x00</td><td rowspan="2">Length = 0x0916</td><td colspan="3">ASIC-1B Configuration</td><td colspan="2"></td><td colspan="3">ASIC-1A Configuration</td><td></td></tr>
<tr><td>ASIC-1B bits: [1163:1155]</td><td>…</td><td>ASIC-1B bits: [10:3]</td><td>ASIC-1B [3:1]</td><td>ASIC-1A bits: [1163:1159]</td><td>ASIC-1A bits: [1158:1151]</td><td>…</td><td>ASIC-1A [6:1]</td><td>Padding = 0b00</td></tr>
<tr><td rowspan="2">1 byte</td><td rowspan="2">2 bytes</td><td>1 byte</td><td>…</td><td>1 byte</td><td colspan="2">1 byte</td><td>1 byte</td><td>…</td><td colspan="2">1 byte</td></tr>
<tr><td colspan="8">291 bytes</td></tr>
<tr><td colspan="10">294 bytes</td></tr>
</table>