Register file structure : regfile_xgs_athena.pdf
Created by jmansill on 2020/04/30 11:52:56
Register file CRC32 : 0x5C4F3EA4

1. Main Parameters

Register file endianness: little endian
Address bus width: 11 bits
Data bus width: 32 bits

2. Memory Map

Section name	Address(es) / Address Ranges	Register name	Access Type
SYSTEM	0x000	TAG	R
	0x004	VERSION	R
	0x008	CAPABILITY	R
	0x00C	SCRATCHPAD	RW
DMA	0x070	CTRL	RW
	0x074	FSTART	RW
	0x078	FSTART_HIGH	RW
	0x07C	FSTART_G	RW
	0x080	FSTART_G_HIGH	RW
	0x084	FSTART_R	RW
	0x088	FSTART_R_HIGH	RW
	0x08C	LINE_PITCH	RW
	0x090	LINE_SIZE	RW
	0x094	CSC	RW
ACQ	0x100	GRAB_CTRL	RW
	0x108	GRAB_STAT	R
	0x110	READOUT_CFG1	RW
	0x114	READOUT_CFG_FRAME_LINE	RW
	0x118	READOUT_CFG2	R
	0x120	READOUT_CFG3	RW
	0x124	READOUT_CFG4	RW
	0x128	EXP_CTRL1	RW
	0x130	EXP_CTRL2	RW
	0x138	EXP_CTRL3	RW
	0x140	TRIGGER_DELAY	RW
	0x148	STROBE_CTRL1	RW
	0x150	STROBE_CTRL2	RW
	0x158	ACQ_SER_CTRL	RW
	0x160	ACQ_SER_ADDATA	RW
	0x168	ACQ_SER_STAT	R
	0x190	SENSOR_CTRL	RW

Section name	Address(es) / Address Ranges	Register name	Access Type
	0x198	SENSOR_STAT	R
	0x19C	SENSOR_SUBSAMPLING	RW
	0x1A4	SENSOR_GAIN_ANA	RW
	0x1A8	SENSOR_ROI_Y_START	RW
	0x1AC	SENSOR_ROI_Y_SIZE	RW
	0x1B0	SENSOR_ROI2_Y_START	RW
	0x1B4	SENSOR_ROI2_Y_SIZE	RW
	0x1B8	SENSOR_M_LINES	RW
	0x1BC	SENSOR_DP_GR	RW
	0x1C0	SENSOR_DP_GB	RW
	0x1C4	SENSOR_DP_R	RW
	0x1C8	SENSOR_DP_B	RW
	0x1E0	DEBUG_PINS	RW
	0x1E8	TRIGGER_MISSED	RW
	0x1F0	SENSOR_FPS	R
	0x2A0	DEBUG	RW
	0x2A8	DEBUG_CNTR1	R
	0x2B8	EXP_FOT	RW
	0x2C0	ACQ_SFNC	RW
DATA	0x300	LUT_CTRL	RW
	0x308	LUT_RB	R
	0x310	WB_MULT1	RW
	0x318	WB_MULT2	RW
	0x320	WB_B_ACC	R
	0x328	WB_G_ACC	R
	0x330	WB_R_ACC	R
	0x338	FPN_ADD	RW
	0x33C	FPN_READ_REG	RW
	0x340, 0x344, ... ,0x35C	FPN_DATA (7:0)	RW
	0x360	FPN_CONTRAST	RW
	0x368	FPN_ACC_ADD	RW
	0x370	FPN_ACC_DATA	R
	0x380	DPC_LIST_CTRL	RW
	0x384	DPC_LIST_DATA	RW
	0x388	DPC_LIST_DATA_RD	R
HISPI	0x400	CTRL	RW
	0x404	STATUS	R

Section: SYSTEM

Address Range: [0x000 - 0x00C]

TAG

Address: section "SYSTEM" base address + 0x000

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

VALUE (23:0)	Tag identifier	
STATIC		
Value at Reset:	0x58544d	
Possible Values:	0x58544D	MTX ASCII string

Address: section "SYSTEM" base address + 0x004

Description:

Revisions

1.3.x : First functionnal revision with a single list of multiple Ethernet frames

1.4.x : Second revision. Implements multiple list of frames

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
MAJOR(7:0)							
15	14	13	12	11	10	9	8
MINOR(7:0)							
7	6	5	4	3	2	1	0
HW(7:0)							

MAJOR (7:0) *STATIC*	
Value at Reset:	0x1

MINOR (7:0) *STATIC*	
Value at Reset:	0x5

HW (7:0) *RO*	

Address: section "SYSTEM" base address + 0x008

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

VALUE (7:0) *STATIC*	
Value at Reset:	0x0

Address: section "SYSTEM" base address + 0x00C

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

VALUE (31:0)	
RW	
Value at Reset:	0x0

Section: DMA

Address Range: [0x070 - 0x0A0]

CTRL — Initial Grab Address Register

Address: section "DMA" base address + 0x000

Description:

Initial Grab Address LOW 32 bits

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

GRAB_QUEUE_EN *RW*		
Value at Reset:	N/A (Non-resettable flip-flop used)	
Possible Values:	0x0	
	0x1	

Address: section "DMA" base address + 0x004

Description:

Initial Grab Address LOW 32 bits

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

VALUE (31:0)	INitial GRAb ADDRess Register	
RW	This is the address in the host ram where the grab engine will start writing pixel data.	
Value at Reset:	N/A (Non-resettable flip-flops used)	
Possible Values:	Any Value	

Address: section "DMA" base address + 0x008

Description:

Initial Grab Address HI 32 bits

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

VALUE (31:0)	INitial GRAb ADDRess Register High	
RW	This is the high 32 bits of the 64-bit addresess in the host ram where the grab engine will start writing pixel data.	
Value at Reset:	N/A (Non-resettable flip-flops used)	
Possible Values:	Any Value	

Address: section "DMA" base address + 0x00C

Description:

Grab Address LOW 32 bits for the Green plane. Only used when grabbing in Planar mode.

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

VALUE (31:0)	GRAb ADDRess Register	
RW	This is the address in the host ram where the grab engine will start writing pixel data.	
Value at Reset:	N/A (Non-resettable flip-flops used)	
Possible Values:	Any Value	

FSTART_G_HIGH — Green Grab Address Register HIGH 32 bits

Address: section "DMA" base address + 0x010

Description:

Green Grab Address HIGH 32 bits

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

VALUE (31:0)	GRAb ADDRess Register High	
RW	This is the high part of the 64-bit addresess in the host ram where the grab engine will start writing pixel data.	
Value at Reset:	N/A (Non-resettable flip-flops used)	
Possible Values:	Any Value	Any value

Address: section "DMA" base address + 0x014

Description:

Grab Address LOW 32 bits for the Red plane. Only used when grabbing in Planar mode.

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

VALUE (31:0)	GRAb ADDRess Register	
RW	This is the address in the host ram where the grab engine will start writing pixel data.	
Value at Reset:	N/A (Non-resettable flip-flops used)	
Possible Values:	Any Value	Any value

Address: section "DMA" base address + 0x018

Description:

Red Grab Address HIGH 32 bits

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

VALUE (31:0)	GRAb ADDRess Register High	
RW	This is the high part of the 64-bit addresess in the host ram where the grab engine will start writing pixel data.	
Value at Reset:	N/A (Non-resettable flip-flops used)	
Possible Values:	Any Value	Any value

Address: section "DMA" base address + 0x01C

Description:

Grab Line Pitch Register

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

VALUE (15:0)	Grab LinePitch
RW	This is the line pitch when writing in ram. It is measured in bytes, not pixels.
Value at Reset:	N/A (Non-resettable flip-flops used)

Address: section "DMA" base address + 0x020

Description:

Host Line Size Register.

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

VALUE (13:0)	Host Line size	
RW	This is the line size when writing in host ram. It is measured in bytes, not pixels. If this register is higher than the actual data provided by the sensor, stray data will be written into host memory. If this register is lower than the data provided by the sensor, image data will be cropped at the end of the line. For backward compatibility, the value of 0 indicates that the FPGA should auto-compute the line sized based on data provided by the sensor interface.	
Value at Reset:	0x0	
Possible Values:	0x1 - 0x3FFF	Written line size in host frame.
	0x0	Auto-compute line size from sensor data.

Address: section "DMA" base address + 0x024

31	30	29	28	27	26	25	24
Reserved					COLOR_SPACE(2:0)		
23	22	21	20	19	18	17	16
DUP_LAST_ LINE	Reserved						
15	14	13	12	11	10	9	8
Reserved						REVERSE_Y	REVERSE_X
7	6	5	4	3	2	1	0
Reserved							

COLOR_SPACE (2:0)		
RW	Output color space used to transfer data to the DMA engine.	
Value at Reset:	0x0	
Possible Values:	0x0	Reserved for Mono sensor operation
	0x1	BGR32
	0x2	YUV 4:2:2 in full range
	0x3	Planar 8-bits
	0x4	Reserved for Y only with color sensor
	0x5	RAW color pixels (8bpp or 10bpp selected with MONO10 regsiter)

DUP_LAST_LINE		
RW	This field is used to enable the duplicate last line feature. When turned on, the datapath will regenerate the last line when it receives the end of frame marker from the acquisition section. The goal of this feature is to compensate for the lost line during the Bayer demosaic processing.	
Value at Reset:	0x0	
Possible Values:	0x0	normal processing
	0x1	last line is duplicated

REVERSE_Y	REVERSE Y	
RW	Reverse readout	
Value at Reset:	0x0	
Possible Values:	0x0	Bottom to top readout
	0x1	Top to bottom readout

REVERSE_X	
RW	
Value at Reset:	0x0

Section: ACQ

Address Range: [0x100 - 0x2CC]

GRAB_CTRL — GRAB ConTRoL Register

Address: section "ACQ" base address + 0x000

Description:

Grag Control Register

31	30	29	28	27	26	25	24
RESET_GRAB	Reserved	GRAB_ROI2_EN	ABORT_GRAB	Reserved			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							TRIGGER_OVERLAP_BUFFn
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TRIGGER_OVERLAP	TRIGGER_ACT(2:0)			Reserved	TRIGGER_SRC(2:0)		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			GRAB_SS	Reserved		BUFFER_ID	GRAB_CMD

RESET_GRAB		
RW	This register resets the entire python_ctrl.	
Value at Reset:	0x0	
Possible Values:	0x0	Reset not active
	0x1	Reset active

GRAB_ROI2_EN		
RW	Enable the second ROI on the frame (KNS). This register is not DB. 1) No Y overlap is allowed 2) Xsize must be the same for the two ROI for the moment(DMA constraint). 3) EOF and SOF in between the two in-frame ROIs will be masked to the DMA. The DMA will see one frame, with the two ROI inside.	
Value at Reset:	0x0	
Possible Values:	0x0	Dual ROI disable
	0x1	Dual ROI enable

ABORT_GRAB	ABORT GRAB	
WO/AutoClr	This is the grab Abort signal, it will reset all the grab queued.	
Possible Values:	0x0	Normal operation
	0x1	Reset Grab

TRIGGER_OVERLAP_BUFFn		
RW	NOT FULLY VALIDATED. DON'T USE. SET IT TO '0'.	
Value at Reset:	0x0	
Possible Values:	0x0	Buffer the trigger received during the dead window in PET mode and execute
	0x1	The trigger will be ignored during dead window in PET mode.

TRIGGER_OVERLAP		
RW	This field enables the trigger overlap. In this mode the exposure and the readout of the sensor can be done in parallel for higher framerates.	
Value at Reset:	0x1	
Possible Values:	0x0	Trigger Overlap disable
	0x1	Trigger Overlap enable (default)

TRIGGER_ACT (2:0)	TRIGGER ACTivation	
RW	This is the trigger activation . This register selects the activation of the trigger when the trigger source is set to Hardware Snapshop mode . This register is Double Buffered, so the trigger activation may change from one grab command to another. In activation Level HI/LO with EXPOSURE_MODE register set to Timed, the camera will be triggered in continuous way if the level of the external trigger remains at the LEVEL programmed in this register. In activation Level HI/LO with EXPOSURE_MODE register set to Trigger Width, the Exposure time will be set by the level of the trigger input. The FPGA exposure regsiters will be ignored. The Dual and Triple slope are not supported in the mode.	
Value at Reset:	0x0	
Possible Values:	0x0	Rising edge
	0x1	Falling edge
	0x2	Rising or Falling edge
	0x3	Level HI
	0x4	Level LO
	0x5	RESERVED
	0x6	RESERVED
	0x7	RESERVED

TRIGGER_SRC (2:0)	TRIGGER SouRCe	
RW	This is the trigger source. This register selects the source of the grab trigger. This register is Double Buffered, so the trigger source may change from one grab command to another. TRIGGER_SRC(1) may be seen as a TRIGGER_STATE by the software driver.	
Value at Reset:	0x0	
Possible Values:	0x0	RESERVED
	0x1	Immediate mode (Continuous)
	0x2	Hardware Snapshop mode
	0x3	Software Snapshot mode
	0x4	SFNC mode (auto trig)

GRAB_SS	GRAB Software Snapshot	
WO/AutoClr	This is the software snapshot register when the trigger source selected is Software Snapshot mode.	
Possible Values:	0x0	Idle
	0x1	Start a grab

BUFFER_ID	
RW	This is the ID of the DMA parameters to associate with this grab command.
Value at Reset:	0x0

GRAB_CMD	GRAB CoMmanD	
WO/AutoClr	This is MIL GRAB command. When the trigger source is set to Immediate mode(Continuous), an exposure sequence will be automatically executed. When the trigger source is set to Software Snapshop mode or Hardware Snapshop mode, GRAB_CMD will act as an ARM. The GRAB_CMD will take around 13 clks to reccord the grab parametters to the SPI fifo. The GRAB_CMD_DONE register may be readed to avoid fifo corruption before sending another Grab command instruction.	
Possible Values:	0x0	Idle
	0x1	Start grab command

GRAB_STAT

Address: section "ACQ" base address + 0x008

31	30	29	28	27	26	25	24
GRAB_CMD_DONE	ABORT_PET	ABORT_DELAI	ABORT_DONE	Reserved			TRIGGER_RDY
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved	ABORT_MNGR_STAT(2:0)			TRIG_MNGR_STAT(3:0)			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	TIMER_MNGR_STAT(2:0)			GRAB_MNGR_STAT(3:0)			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	GRAB_FOT	GRAB_READOUT	GRAB_EXPOSURE	Reserved	GRAB_PENDING	GRAB_ACTIVE	GRAB_IDLE

GRAB_CMD_DONE	GRAB CoMmanD DONE	
RO	The GRAB_CMD will take around 13 clks to reccord the grab parametters to the SPI fifo. This register may be readed to avoid fifo corruption before sending another Grab command instruction.	
Possible Values:	0x0	Grab Command in process
	0x1	Grab command idle

ABORT_PET	ABORT during PET	
RO	This is the ABORT PET flag. It is set to '1' when an abort is detected in the PETengin phase of the trigger. It is set back to '0' when ABORT_DONE is set to '1'.	
Possible Values:	0x0	Abort in PET Phase idle
	0x1	Abort in PET Phase active

ABORT_DELAI		
RO	This is the ABORT DELAI flag. It is set to '1' when an abort is detected in the delai phase of the trigger. It is set back to '0' when ABORT_DONE is set to '1'.	
Possible Values:	0x0	Abort in Delai Phase idle
	0x1	Abort in Delai Phase active

ABORT_DONE	ABORT is DONE	
RO	This read-only field indicates the RESET_GRAB command status. If 0, an abort sequence is executing.	
Possible Values:	0x0	Abort sequence not finished yet
	0x1	Abort DONE, or not started (reset value)

TRIGGER_RDY	
RO	

ABORT_MNGR_STAT (2:0)	
RO	DEBUG ABORT MANAGER STATE MACHINE

TRIG_MNGR_STAT (3:0)	
RO	DEBUG TRIGGER MANAGER STATE MACHINE

TIMER_MNGR_STAT (2:0)	
RO	DEBUG TIMER MANAGER STATE MACHINE

GRAB_MNGR_STAT (3:0)	
RO	DEBUG GRAB MANAGER STATE MACHINE

GRAB_FOT	GRAB Field Overhead Time	
RO	This is the sensor FOT (Field Overhead Time).	
Possible Values:	0x0	Not in FOT
	0x1	In FOT

GRAB_READOUT	
RO	This is the sensor readout status. It goes to '1' on the SO_FOT and goes to '0' when the datapath decoder decodes the end of frame.

GRAB_EXPOSURE		
RO	This is the sensor integration status	
Possible Values:	0x0	Idle
	0x1	Integrating

GRAB_PENDING		
RO	Grab pending status. When this register is set to one, a second grab command is queued in the fpga.	
Possible Values:	0x0	No grab pending
	0x1	Grab pending

GRAB_ACTIVE	
RO	Grab active status. When this register is set to one, at least one grab command has been received.

GRAB_IDLE		
RO	GRAB IDLE status. When this register is set to '1', The grab engin is in idle state.	
Possible Values:	0x0	Grab is in proccess
	0x1	Grab is Idle

READOUT_CFG1

Address: section "ACQ" base address + 0x010

31	30	29	28	27	26	25	24
Reserved			FOT_LENGTH_LINE(4:0)				
23	22	21	20	19	18	17	16
Reserved							EO_FOT_SEL
15	14	13	12	11	10	9	8
FOT_LENGTH(15:8)							
7	6	5	4	3	2	1	0
FOT_LENGTH(7:0)							

FOT_LENGTH_LINE (4:0)	Frame Overhead Time LENGTH LINE	
RW	This is the length of the Frame Overhead Time in line_time unit.	
Value at Reset:	0x0	
Possible Values:	Any Value	Any 16 bit value

EO_FOT_SEL	
RW	This selector selects who will generate the EO_FOT in the controller. When select 0, the EO_FOT is the falling edge detection of the monitor FOT. When select 1, the EO_FOT will be generated inside the controller with programmed FOT_LENGTH.
Value at Reset:	0x0

FOT_LENGTH (15:0)	Frame Overhead Time LENGTH	
RW	This is the length of the Frame Overhead Time. This register is defined as number of lines. It is used when EO_FOT_SEL is set to 1.	
Value at Reset:	0x0	
Possible Values:	Any Value	Any 16 bit value

READOUT_CFG_FRAME_LINE

Address: section "ACQ" base address + 0x014

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

DUMMY_LINES (7:0)	
RW	Number of lines to add in the readout (to debug XGS)
Value at Reset:	0x0

CURR_FRAME_LINES (12:0)	
RO	Current number of lines in the readout calculated by the XGS controller (without FOT).

READOUT_CFG2

Address: section "ACQ" base address + 0x018

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

READOUT_LENGTH (28:0)		
RO	This is the readout length register. This register is calculated by the FPGA in the IRIS4 projectand gives the readout lenght without the FOT. This register will depend on the ROI, and Subsampling mode. It is used in the PET engin calculations. In Sys_Clock domain.	
Possible Values:	Any Value	Any 24 bits value

READOUT_CFG3

Address: section "ACQ" base address + 0x020

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							KEEP_OUT_TRIG_ENA
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
LINE_TIME(15:8)							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
LINE_TIME(7:0)							

KEEP_OUT_TRIG_ENA	
RW	KEEPOUT zone TRIGger ENAble. When this register is enabled, then the trigger output will be synchronized with the line_int(monitor2) signal from the XGS sensor. To configure this keep out zone, use register READOUT_CFG4.
Value at Reset:	0x0

LINE_TIME (15:0)	LINE TIME	
RW	This register definel the length of one line of the sensor. It includes blanking and valid time . Line Time Unit is SENSOR Clock Cycles	
Value at Reset:	0x16e	
Possible Values:	Any Value	between 1 and 255

READOUT_CFG4

Address: section "ACQ" base address + 0x024

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

KEEP_OUT_TRIG_END (15:0)	
RW	During the line time, this register indicates the end of the trigger keep-out zone.
Value at Reset:	0x16d

KEEP_OUT_TRIG_START (15:0)	
RW	During the line time, this register indicates the start of the trigger keep-out zone.
Value at Reset:	0x16e

EXP_CTRL1

Address: section "ACQ" base address + 0x028

31	30	29	28	27	26	25	24
Reserved			EXPOSURE_ LEV_MODE	EXPOSURE_SS(27:24)			
23	22	21	20	19	18	17	16
EXPOSURE_SS(23:16)							
15	14	13	12	11	10	9	8
EXPOSURE_SS(15:8)							
7	6	5	4	3	2	1	0
EXPOSURE_SS(7:0)							

EXPOSURE_LEV_MODE	EXPOSURE LEVel MODE	
RW	This is the exposure level mode selector. When selecting the TRIGGER ACTIVATION = Level Mode, this register selects the exposure method used. When this register is set to '0' the timed mode is selected; Register EXPOSURE_SS is used for the exposure time. When this register is set to '1' the external trigger width is used for the exposure time.	
Value at Reset:	0x0	
Possible Values:	0x0	Timed Mode
	0x1	Trigger Width

EXPOSURE_SS (27:0)	EXPOSURE Single Slope	
RW	This is the total exposure time in single/dual/triple slope mode. This register is double buffered.	
Value at Reset:	0x0	
Possible Values:	Any Value	Any 28 bits value

EXP_CTRL2

Address: section "ACQ" base address + 0x030

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

EXPOSURE_DS (27:0)	EXPOSURE Dual	
RW	This is a new 3d profiler feature We will be able to program upto 3 diferent exposure times (using unused multiSlope registers) Then we will be able to sequence those exposure times . Selection is made with input exposure_select.	
Value at Reset:	0x0	
Possible Values:	Any Value	Any 28 bits value

EXP_CTRL3

Address: section "ACQ" base address + 0x038

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

EXPOSURE_TS (27:0)	EXPOSURE Tripple	
RW	This is a new 3d profiler feature We will be able to program upto 3 diferent exposure times (using unused multiSlope registers) Then we will be able to sequence those exposure times . Selection is made with input exposure_select.	
Value at Reset:	0x0	
Possible Values:	Any Value	Any 28 bits value

TRIGGER_DELAY

Address: section "ACQ" base address + 0x040

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

TRIGGER_DELAY (27:0)	TRIGGER DELAY	
RW	This is the trigger delay. This trigger delay can be applied to HW(Only edge mode), SW and Continuous mode. In HW level mode, the trigger cannot be delayed, since the level time represents the exposure time. This register is double buffered	
Value at Reset:	0x0	
Possible Values:	Any Value	Any 28 bits value

STROBE_CTRL1

Address: section "ACQ" base address + 0x048

31	30	29	28	27	26	25	24
STROBE_E	Reserved		STROBE_POL	STROBE_START(27:24)			
23	22	21	20	19	18	17	16
STROBE_START(23:16)							
15	14	13	12	11	10	9	8
STROBE_START(15:8)							
7	6	5	4	3	2	1	0
STROBE_START(7:0)							

STROBE_E	STROBE Enable	
RW	This register enables the strobe logic. For Nexis 3 systems, to enable STROBE_A signal, STROBE_E and STROBE_A_EN must be enabled. For Nexis 3 systems, to enable STROBE_B signal, STROBE_E and STROBE_B_EN must be enabled. For Nexis 3 systems, STROBE_A and STROBE B can be activated at the same time, in this case the two strobes will be the same as they share the same programmation. This register is double buffered	
Value at Reset:	0x0	
Possible Values:	0x0	Strobe disabled
	0x1	Strobe enabled

STROBE_POL	STROBE POLarity	
RW	This is the strobe polarity at the pin of the FPGA only for GTR systems. For NEXIS3 systems use register ANPUT\IO\IO_OUT_POL\OUTx_POL This register is not double buffered.	
Value at Reset:	0x0	
Possible Values:	0x0	Active high strobe
	0x1	Active low strobe

STROBE_START (27:0)	STROBE START	
RW	This is the strobe start location. This location depends on the Strobe Mode used. In Strobe Mode='0', the start of the strobe is situated during the exposure time. In Strobe Mode='1', the start of the strobe is situated during the trigger delay. This register is double buffered	
Value at Reset:	0x0	
Possible Values:	Any Value	Any 28 bits value

STROBE_CTRL2

Address: section "ACQ" base address + 0x050

31	30	29	28	27	26	25	24
STROBE_MODE	Reserved	STROBE_B_EN	STROBE_A_EN	STROBE_END(27:24)			
23	22	21	20	19	18	17	16
STROBE_END(23:16)							
15	14	13	12	11	10	9	8
STROBE_END(15:8)							
7	6	5	4	3	2	1	0
STROBE_END(7:0)							

STROBE_MODE	STROBE MODE	
RW	This register selects the location of the Strobe Start. When this register is set to 0, the STROBE_START register is located during the exposure timer. When this register is set to 1, the STROBE_START register is located during the trigger delay timer. In HW level mode the strobe mode must be set to STROBE MODE=0 since the trigger cannot be delayed. This register is double buffered	
Value at Reset:	0x0	
Possible Values:	0x0	Strobe start during exposure
	0x1	Strobe start during trigger delay

STROBE_B_EN	STROBE phase B ENable	
RW	This field enables the generation of STROBE_B signal, for a NEXIS 3 system. This register is double buffered to support back2back mode in nexis systems.	
Value at Reset:	0x0	
Possible Values:	0x0	Enable Strobe B
	0x1	Disable Strobe B

STROBE_A_EN	STROBE phase A ENable	
RW	This field enables the generation of STROBE_A signal(Default strobe), for a NEXIS 3 system. This register is double buffered to support back2back mode in nexis systems.	
Value at Reset:	0x1	
Possible Values:	0x0	Enable Strobe A (default strobe)
	0x1	Disable Strobe A

STROBE_END (27:0)	STROBE END	
RW	This is the strobe end location. This location does not depend on the Strobe Mode used. This register is double buffered	
Value at Reset:	0xfffffff	
Possible Values:	Any Value	Any 28 bits value

Address: section "ACQ" base address + 0x058

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							SER_RWn
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						SER_CMD(1:0)	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			SER_RF_SS	Reserved			SER_WF_SS

SER_RWn	SERial Read/Writen	
RW	This register configures the type of the serial access to the CMOS sensor	
Value at Reset:	0x1	
Possible Values:	0x0	Write access
	0x1	Read access

SER_CMD (1:0)	SERial CoMmand	
RW	This is the type of command sent to the serial fifo. To access the Sensor, write SER_WF_SS=1 with SER_CMD=0x0, with the parametters: SER_WRn, SER_ADD(8:0) and SER_DAT(15:0). To insert a timer between fifo commands, write SER_WF_SS=1 with SER_CMD=0x1, with the parameter: SER_DAT(15:0). The value of the timer inserted is calculated with the following formula: Timer= SER_DAT(15:0)*1024*SYS_PERIOD, SYS_PERIOD is 1/62.5mhz. The granularity of the timer is 16.384us To insert a Stop separator command, write SER_WF_SS=1 with SER_CMD=0x3. When the read logic encounter this command, it will stop read from the fifo until a new SER_RF_SS is received.	
Value at Reset:	0x0	
Possible Values:	0x0	CMOS sensor access COMMAND
	0x1	Insert timer COMMAND
	0x2	STOP separator COMMAND
	0x3	RESERVED

SER_RF_SS	SERial Read Fifo SnapShot	
WO/AutoClr	This is the read fifo snapshot. When the read fifo logic receives this snapshot, it will read all the fifo comands until a STOP separator command is read or Empty fifo is detected.	
Possible Values:	0x0	Idle
	0x1	Start Read FIFO

SER_WF_SS	SERial Write Fifo SnapShot	
WO/AutoClr	When the system toggle this bit, the address, data and command are wrote to the command fifo. This fifo can contain the entire dcf, so the driver will not need to pool the status bit. This is a auto reset bit register, so after the driver write one, the bit will be auto reset to 0. To start the FIFO read logic write '1' to regsiter SER_RF_SS.	
Possible Values:	0x0	Idle
	0x1	Write a command to the FIFO

Address: section "ACQ" base address + 0x060

31	30	29	28	27	26	25	24
SER_DAT(15:8)							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SER_DAT(7:0)							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	SER_ADD(14:8)						
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SER_ADD(7:0)							

SER_DAT (15:0)	SERial interface DATa	
RW	This is the write data to be send to the CMOS sensor by the serial interface, or the config data to a TIMER command or to a POWER sequence command. See register SER_CMD.	
Value at Reset:	0x0	
Possible Values:	Any Value	Any 16 bits value

SER_ADD (14:0)	SERial interface ADDress	
RW	This is the read/write address of the register in the CMOS sensor.	
Value at Reset:	0x0	
Possible Values:	Any Value	Any 9 bits value

Address: section "ACQ" base address + 0x068

31	30	29	28	27	26	25	24
Reserved							SER_FIFO_EMPTY
23	22	21	20	19	18	17	16
Reserved							SER_BUSY
15	14	13	12	11	10	9	8
SER_DAT_R(15:8)							
7	6	5	4	3	2	1	0
SER_DAT_R(7:0)							

SER_FIFO_EMPTY	SERial FIFO EMPTY
RO	This is the EMPTY flag of the xilinx fifo, when '1' there are no pending operations in the fifo.

SER_BUSY	SERial BUSY	
RO	This is the BUSY status of the FIFO read logic. The flag will be set to '1' when the SER_RF_SS is set to '1'. It will be reseted to '0' when the read logic will decode a STOP separator command or when the FIFO will be empty.	
Possible Values:	0x0	FIFO read logic is idle
	0x1	FIFO read logic is runnning

SER_DAT_R (15:0)	SERial interface DATa Read	
RO	This is the data read from CMOS sensor.	
Possible Values:	Any Value	Any 16 bits value

Address: section "ACQ" base address + 0x090

31	30	29	28	27	26	25	24
Reserved							SENSOR_REFRESH_TEMP
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							SENSOR_POWERDOWN
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							SENSOR_COLOR
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			SENSOR_REG_UPTATE	Reserved		SENSOR_RESETN	SENSOR_POWERUP

SENSOR_REFRESH_TEMP	SENSOR REFRESH TEMPerature	
WO/AutoClr	This register starts a sensor temperature read on the serial interface of the Python sensor. The temperature value readed will be available on field SENSOR_TEMP when field SENSOR_TEMP_VALID is set to '1'. [Pas utilise pour le moment dans IRIS4]	
Possible Values:	0x0	Idle
	0x1	Starts a Temperature read on Python SPI interface

SENSOR_POWERDOWN	
WO/AutoClr	After a PowerUp sequence(SESOR_POWERUP_DONE=1), successfull or not, this register can reset the clock oscillator and enable the reset to the sensor. This power down don't do power sequencing.

SENSOR_COLOR	SENSOR COLOR	
RW	This register informs the datapath logic that a color sensor is used. This information is needed for the remapper logic.	
Value at Reset:	0x0	
Possible Values:	0x0	Monochrone sensor
	0x1	Color sensor

SENSOR_REG_UPTATE	SENSOR REGister UPDATE	
RW	By setting this bit to 1, the SENSOR CONTROLLER WILL UPDATE the programed CMOS sensor registers at the beginning of each grab.	
Value at Reset:	0x1	
Possible Values:	0x0	Do not update registers
	0x1	Update registers

SENSOR_RESETN	SENSOR RESET Not	
RW	After a successfull PowerUP sequence, writing this field to '0' reset the Python CMOS sensor.	
Value at Reset:	0x1	
Possible Values:	0x0	Reset the sensor after a successfull powerUP
	0x1	Nothing

SENSOR_POWERUP		
WO/AutoClr	This register Enables the clk oscillator and removes the reset from the sensor.	
Possible Values:	0x0	idle
	0x1	Start the power sequence

Address: section "ACQ" base address + 0x098

31	30	29	28	27	26	25	24
SENSOR_TEMP(7:0)							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SENSOR_TEMP_VALID	Reserved						SENSOR_POWERDOWN
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		SENSOR_RESETN	SENSOR_OSC_EN	Reserved			SENSOR_VCC_PG
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						SENSOR_POWERUP_STAT	SENSOR_POWERUP_DONE

SENSOR_TEMP (7:0)		
RO	This register gives the Temperature of the Python sensor after a SENSOR_REFRESH_TEMP snapshot. The field SENSOR_TEMP_VALID indicates when the SENSOR_TEMP value is valid. [Pas utilise pour le moment dans IRIS4]	
Possible Values:	Any Value	

SENSOR_TEMP_VALID	SENSOR TEMPerature VALID	
RO	This field indicates that the field SENSOR_TEMP have valid temperature after a SENSOR_REFRESH_TEMP snapshot. [Pas utilise pour le moment dans IRIS4]	
Possible Values:	0x0	SENSOR_TEMPERATURE register is not valid
	0x1	SENSOR_TEMPERATURE register is valid

SENSOR_POWERDOWN		
RO	This field indicates that the sensor is in powerdown state.	
Possible Values:	0x0	Not in powerdown state
	0x1	Powerdown

SENSOR_RESETN	SENSOR RESET N	
RO	This is the sensor RESETN status.	
Possible Values:	0x0	In reset state
	0x1	Not in reset

SENSOR_OSC_EN	SENSOR OSCILLATOR ENable	
RO	This is the sensor oscillator enable status.	
Possible Values:	0x0	Disable
	0x1	Enable

SENSOR_VCC_PG	SENSOR supply VCC Power Good	
RO	This is the VCC Power Good status (generated by external HW). [TO BE DELETED, waiting for ON SEMI INFORMATION]	
Possible Values:	0x0	Disable
	0x1	Enable

SENSOR_POWERUP_STAT		
RO	When a powerup sequence is finish, this register indicates the result of the POWERUP sequence.	
Possible Values:	0x0	PowerUp sequence fail
	0x1	PowerUp sequence success

SENSOR_POWERUP_DONE		
RO	This register indicates that the POWERUP sequence is finish. Read register SENSOR_POWERUP_STAT to see the result.	
Possible Values:	0x0	PowerUp sequence not started
	0x1	PowerUp sequence finish

SENSOR_SUBSAMPLING

Address: section "ACQ" base address + 0x09C

Description:
SENSOR ADDRESS

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
reserved1(11:4)							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
reserved1(3:0)				ACTIVE_SUBSAMPLING_Y	reserved0	M_SUBSAMPLING_Y	SUBSAMPLING_X

reserved1 (11:0)		
STATIC		
Value at Reset:	0x0	

ACTIVE_SUBSAMPLING_Y		
RW	Subsampling (Row) for ROI Configurations	
Value at Reset:	0x0	
Possible Values:	0x0	
	0x1	

reserved0		
STATIC		
Value at Reset:	0x0	
Possible Values:	0x0	Idle
	0x1	Enable

M_SUBSAMPLING_Y		
RW	Subsampling (Row) for M Region	
Value at Reset:	0x0	
Possible Values:	0x0	
	0x1	

SUBSAMPLING_X		
RW	Readout in Column Subsampling Mode	
Value at Reset:	0x0	
Possible Values:	0x0	
	0x1	

SENSOR_GAIN_ANA

Address: section "ACQ" base address + 0x0A4

Description:

SENSOR ADDRESS 204 DEC

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
reserved1(4:0)					ANALOG_GAIN(2:0)		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
reserved0(7:0)							

reserved1 (4:0) *STATIC*	
Value at Reset:	0x0

ANALOG_GAIN (2:0) *RW*		
Value at Reset:	0x1	
Possible Values:	0x1	1x
	0x3	2x
	0x7	4x

reserved0 (7:0) *STATIC*	
Value at Reset:	0x0

SENSOR_ROI_Y_START

Address: section "ACQ" base address + 0x0A8

Description:
SENSOR ADDRESS

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
reserved(5:0)						Y_START(9:8)	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Y_START(7:0)							

reserved (5:0)	
STATIC	
Value at Reset:	0x0

Y_START (9:0)	Y START
RW	Y Start in Kernel size (Kernel is 4 lines)
Value at Reset:	0x0

SENSOR_ROI_Y_SIZE

Address: section "ACQ" base address + 0x0AC

Description:

SENSOR ADDRESS

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

reserved (5:0)	
STATIC	
Value at Reset:	0x0

Y_SIZE (9:0)	Y SIZE
RW	Y SIZE in Kernel size (Kernel is 4 lines)
Value at Reset:	0x302

SENSOR_ROI2_Y_START

Address: section "ACQ" base address + 0x0B0

Description:
SENSOR ADDRESS

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
reserved(5:0)						Y_START(9:8)	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Y_START(7:0)							

reserved (5:0)	
STATIC	
Value at Reset:	0x0

Y_START (9:0)	Y START
RW	Y Start in Kernel size (Kernel is 4 lines)
Value at Reset:	0x0

SENSOR_ROI2_Y_SIZE

Address: section "ACQ" base address + 0x0B4

Description:
SENSOR ADDRESS

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

reserved (5:0)	
STATIC	
Value at Reset:	0x0

Y_SIZE (9:0)	Y SIZE
RW	Y SIZE in Kernel size (Kernel is 4 lines)
Value at Reset:	0x302

SENSOR_M_LINES

Address: section "ACQ" base address + 0x0B8

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	M_SUPPRESSED(4:0)					M_LINES_SENSOR(9:8)	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
M_LINES_SENSOR(7:0)							

M_SUPPRESSED (4:0)	
RW	Suppress the Readout of Initial Lines in the M Region
Value at Reset:	0x0

M_LINES_SENSOR (9:0)	
RW	Number of Lines to Readout from M Region in Context 0 Unit is #lines Total number of Black lines = M_LINES Total number of Black lines transfered as valid Black lines= M_LINES-M_SUPRESSED
Value at Reset:	0x8

SENSOR_DP_GR

Address: section "ACQ" base address + 0x0BC

Description:

Sensor Analog data pedestal for Gr pixels (Black offset)

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
reserved(3:0)				DP_OFFSET_GR(11:8)			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DP_OFFSET_GR(7:0)							

reserved (3:0)	
STATIC	
Value at Reset:	0x0

DP_OFFSET_GR (11:0)	
RW	Sensor Analog data pedestal for Gr pixels (Black offset)
Value at Reset:	0x100

SENSOR_DP_GB

Address: section "ACQ" base address + 0x0C0

Description:

Sensor Analog data pedestal for Gb pixels (Black offset)

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
reserved(3:0)				DP_OFFSET_GB(11:8)			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DP_OFFSET_GB(7:0)							

reserved (3:0)	
STATIC	
Value at Reset:	0x0

DP_OFFSET_GB (11:0)	
RW	Sensor Analog data pedestal for Gb pixels (Black offset)
Value at Reset:	0x100

SENSOR_DP_R

Address: section "ACQ" base address + 0x0C4

Description:

Sensor Analog data pedestal for R pixels (Black offset)

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
reserved(3:0)				DP_OFFSET_R(11:8)			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DP_OFFSET_R(7:0)							

reserved (3:0)	
STATIC	
Value at Reset:	0x0

DP_OFFSET_R (11:0)	
RW	Sensor Analog data pedestal for R pixels (Black offset)
Value at Reset:	0x100

SENSOR_DP_B

Address: section "ACQ" base address + 0x0C8

Description:

Sensor Analog data pedestal for B pixels (Black offset)

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

reserved (3:0)	
STATIC	
Value at Reset:	0x0

DP_OFFSET_B (11:0)	
RW	Sensor Analog data pedestal for B pixels (Black offset)
Value at Reset:	0x100

DEBUG_PINS

Address: section "ACQ" base address + 0x0E0

31	30	29	28	27	26	25	24
Reserved			Debug3_sel(4:0)				
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved			Debug2_sel(4:0)				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			Debug1_sel(4:0)				
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			Debug0_sel(4:0)				

Debug3_sel (4:0)	

RW	debug_vector(0x0) <= python_monitor0;
	debug_vector(0x1) <= python_monitor1;
	debug_vector(0x2) <= grab_mngr_trig_rdy;
	debug_vector(0x3) <= curr_trig0;
	debug_vector(0x4) <= strobe;
	debug_vector(0x5) <= python_exposure;
	debug_vector(0x6) <= FOT;
	debug_vector(0x7) <= readout;
	debug_vector(0x8) <= readout_stateD;
	debug_vector(0x9) <= ext_trig;
	debug_vector(0xa) <= REGFILE.ACQ.GRAB_CTRL.GRAB_CMD;
	debug_vector(0xb) <= REGFILE.ACQ.GRAB_CTRL.GRAB_SS;
	debug_vector(0xc)<= grab_mngr_trig;
	debug_vector(0xd) <= grab_mngr_trig_rdy;
	debug_vector(0xe) <= grab_pending;
	debug_vector(0xf) <= grab_active;
	debug_vector(0x10) <= DEC_DATA_EN;
	debug_vector(0x11) <= DEC_SOL;
	debug_vector(0x12) <= DEC_SOF;
	debug_vector(0x13) <= DEC_EOL;
	debug_vector(0x14) <= DEC_EOF;
	debug_vector(0x15) <= DEC_CRC;
	debug_vector(0x16) <= DEC_TRAIN;
	debug_vector(0x17) <= fpnprnu_corr_sof;
	debug_vector(0x18) <= fpnprnu_corr_sol;
	debug_vector(0x19) <= fpnprnu_corr_data_val;
	debug_vector(0x1a) <= fpnprnu_corr_eol;
	debug_vector(0x1b) <= fpnprnu_corr_eof;
	debug_vector(0x1c) <= python_ssn_int;
	debug_vector(0x1d) <= debug_lvds(0);
	debug_vector(0x1e) <= debug_lvds(1);
	debug_vector(0x1f) <= 'Z';

Value at Reset:	0x1f

Debug2_sel (4:0)	

RW	debug_vector(0x0) <= python_monitor0;
	debug_vector(0x1) <= python_monitor1;
	debug_vector(0x2) <= grab_mngr_trig_rdy;
	debug_vector(0x3) <= curr_trig0;
	debug_vector(0x4) <= strobe;
	debug_vector(0x5) <= python_exposure;
	debug_vector(0x6) <= FOT;
	debug_vector(0x7) <= readout;
	debug_vector(0x8) <= readout_stateD;
	debug_vector(0x9) <= ext_trig;
	debug_vector(0xa) <= REGFILE.ACQ.GRAB_CTRL.GRAB_CMD;
	debug_vector(0xb) <= REGFILE.ACQ.GRAB_CTRL.GRAB_SS;
	debug_vector(0xc)<= grab_mngr_trig;
	debug_vector(0xd) <= grab_mngr_trig_rdy;
	debug_vector(0xe) <= grab_pending;
	debug_vector(0xf) <= grab_active;
	debug_vector(0x10) <= DEC_DATA_EN;
	debug_vector(0x11) <= DEC_SOL;
	debug_vector(0x12) <= DEC_SOF;
	debug_vector(0x13) <= DEC_EOL;
	debug_vector(0x14) <= DEC_EOF;
	debug_vector(0x15) <= DEC_CRC;
	debug_vector(0x16) <= DEC_TRAIN;
	debug_vector(0x17) <= fpnprnu_corr_sof;
	debug_vector(0x18) <= fpnprnu_corr_sol;
	debug_vector(0x19) <= fpnprnu_corr_data_val;
	debug_vector(0x1a) <= fpnprnu_corr_eol;
	debug_vector(0x1b) <= fpnprnu_corr_eof;
	debug_vector(0x1c) <= python_ssn_int;
	debug_vector(0x1d) <= debug_lvds(0);
	debug_vector(0x1e) <= debug_lvds(1);
	debug_vector(0x1f) <= 'Z';

Value at Reset:	0x1f

Debug1_sel (4:0)	

RW	debug_vector(0x0) <= python_monitor0;
	debug_vector(0x1) <= python_monitor1;
	debug_vector(0x2) <= grab_mngr_trig_rdy;
	debug_vector(0x3) <= curr_trig0;
	debug_vector(0x4) <= strobe;
	debug_vector(0x5) <= python_exposure;
	debug_vector(0x6) <= FOT;
	debug_vector(0x7) <= readout;
	debug_vector(0x8) <= readout_stateD;
	debug_vector(0x9) <= ext_trig;
	debug_vector(0xa) <= REGFILE.ACQ.GRAB_CTRL.GRAB_CMD;
	debug_vector(0xb) <= REGFILE.ACQ.GRAB_CTRL.GRAB_SS;
	debug_vector(0xc)<= grab_mngr_trig;
	debug_vector(0xd) <= grab_mngr_trig_rdy;
	debug_vector(0xe) <= grab_pending;
	debug_vector(0xf) <= grab_active;
	debug_vector(0x10) <= DEC_DATA_EN;
	debug_vector(0x11) <= DEC_SOL;
	debug_vector(0x12) <= DEC_SOF;
	debug_vector(0x13) <= DEC_EOL;
	debug_vector(0x14) <= DEC_EOF;
	debug_vector(0x15) <= DEC_CRC;
	debug_vector(0x16) <= DEC_TRAIN;
	debug_vector(0x17) <= fpnprnu_corr_sof;
	debug_vector(0x18) <= fpnprnu_corr_sol;
	debug_vector(0x19) <= fpnprnu_corr_data_val;
	debug_vector(0x1a) <= fpnprnu_corr_eol;
	debug_vector(0x1b) <= fpnprnu_corr_eof;
	debug_vector(0x1c) <= python_ssn_int;
	debug_vector(0x1d) <= debug_lvds(0);
	debug_vector(0x1e) <= debug_lvds(1);
	debug_vector(0x1f) <= 'Z';

Value at Reset:	0x1f

Debug0_sel (4:0)	

RW	debug_vector(0x0) <= python_monitor0;
	debug_vector(0x1) <= python_monitor1;
	debug_vector(0x2) <= grab_mngr_trig_rdy;
	debug_vector(0x3) <= curr_trig0;
	debug_vector(0x4) <= strobe;
	debug_vector(0x5) <= python_exposure;
	debug_vector(0x6) <= FOT;
	debug_vector(0x7) <= readout;
	debug_vector(0x8) <= readout_stateD;
	debug_vector(0x9) <= ext_trig;
	debug_vector(0xa) <= REGFILE.ACQ.GRAB_CTRL.GRAB_CMD;
	debug_vector(0xb) <= REGFILE.ACQ.GRAB_CTRL.GRAB_SS;
	debug_vector(0xc)<= grab_mngr_trig;
	debug_vector(0xd) <= grab_mngr_trig_rdy;
	debug_vector(0xe) <= grab_pending;
	debug_vector(0xf) <= grab_active;
	debug_vector(0x10) <= DEC_DATA_EN;
	debug_vector(0x11) <= DEC_SOL;
	debug_vector(0x12) <= DEC_SOF;
	debug_vector(0x13) <= DEC_EOL;
	debug_vector(0x14) <= DEC_EOF;
	debug_vector(0x15) <= DEC_CRC;
	debug_vector(0x16) <= DEC_TRAIN;
	debug_vector(0x17) <= fpnprnu_corr_sof;
	debug_vector(0x18) <= fpnprnu_corr_sol;
	debug_vector(0x19) <= fpnprnu_corr_data_val;
	debug_vector(0x1a) <= fpnprnu_corr_eol;
	debug_vector(0x1b) <= fpnprnu_corr_eof;
	debug_vector(0x1c) <= python_ssn_int;
	debug_vector(0x1d) <= debug_lvds(0);
	debug_vector(0x1e) <= debug_lvds(1);
	debug_vector(0x1f) <= 'Z';

Value at Reset:	0x1f

TRIGGER_MISSED

Address: section "ACQ" base address + 0x0E8

31	30	29	28	27	26	25	24
Reserved			TRIGGER_MISSED_RST	Reserved			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TRIGGER_MISSED_CNTR(15:8)							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TRIGGER_MISSED_CNTR(7:0)							

TRIGGER_MISSED_RST	TRIGGER MISSED ReSeT	
WO/AutoClr	This is the trigger missed reset.	
Possible Values:	0x1	Reset the Trigger counter reset

TRIGGER_MISSED_CNTR (15:0)	TRIGGER MISSED CouNTeR	
RO	This is the number of trigger missed detected.	
Possible Values:	Any Value	

SENSOR_FPS

Address: section "ACQ" base address + 0x0F0

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

SENSOR_FPS (15:0)	SENSOR Frame Per Second
RO	This is the number of frames received in 1 second interval. This register can count up to 64k frame/s. This counter counts on SO_FOT event.

DEBUG

Address: section "ACQ" base address + 0x1A0

31	30	29	28	27	26	25	24
Reserved			DEBUG_RST_CNTR	Reserved			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					LED_TEST_COLOR(1:0)		LED_TEST

DEBUG_RST_CNTR *RW*	This register clears the debug cntrs	
Value at Reset:	0x1	
Possible Values:	0x0	
	0x1	Reset counters

LED_TEST_COLOR (1:0) *RW*		
Value at Reset:	0x0	
Possible Values:	0x0	The LED is OFF
	0x1	The LED is GREEN
	0x2	The LED is RED
	0x3	The LED is ORANGE

LED_TEST *RW*	This register will put the LED status in test mode. The test mode is controlled by LED_TEST_COLOR	
Value at Reset:	0x0	
Possible Values:	0x0	The LED is in user mode.
	0x1	The LED is in test mode.

DEBUG_CNTR1

Address: section "ACQ" base address + 0x1A8

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

SENSOR_FRAME_DURATION (27:0)		
RO	This is the time between the last 2 EOF received(in sys clock domain). This register can count up to 4.29 seconds. It can be used to predict sensor framerate or to verify sync between 3D profiler heads. This feature is enabled by setting register regfile.ACQ.DEBUG.DEBUG_RST_CNTR to 0.	
Possible Values:	Any Value	Any 28 bits value

EXP_FOT

Address: section "ACQ" base address + 0x1B8

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							EXP_FOT
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				EXP_FOT_TIME(11:8)			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EXP_FOT_TIME(7:0)							

EXP_FOT	EXPosure during FOT	
RW	When set to '1' this register, the output exposure and strobe signals will take into account the exposure in the FOT of the frame. This timing must be programmed in register EXP_FOT_TIME. This timing must be calculated from the OnSemi setting files .	
Value at Reset:	0x1	
Possible Values:	0x0	Disable exposure during FOT in output exposure signal and Strobe
	0x1	Enable exposure during FOT in output exposure signal and Strobe

EXP_FOT_TIME (11:0)	EXPosure during FOT TIME
RW	This is the time of the exposure during the FOT. This timing must be calculated from the OnSemi setting files . From DCF v1.2, for all LVDS modes : P5000 & P2000 EXP_FOT=40.666us, program value 0x9ee P1300 & P500 & P300 EXP_FOT=27.333us, program value 0x6ac
Value at Reset:	0x9ee

ACQ_SFNC

Address: section "ACQ" base address + 0x1C0

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

RELOAD_GRAB_PARAMS *RW*	This register is not used for the moment. It may be used in the future to reload the exposure time	
Value at Reset:	0x1	
Possible Values:	0x0	
	0x1	

Section: DATA

Address Range: [0x300 - 0x388]

LUT_CTRL

Address: section "DATA" base address + 0x000

31	30	29	28	27	26	25	24
LUT_BYPASS	Reserved	LUT_PALETTE_USE	LUT_PALETTE_W	Reserved		LUT_DATA_W(9:8)	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
LUT_DATA_W(7:0)							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	LUT_SEL(2:0)			LUT_WRN	LUT_SS	LUT_ADD(9:8)	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
LUT_ADD(7:0)							

LUT_BYPASS	LUT BYPASS	
RW	When set this register to '1', the LUT logic will not be used, and the 10 bits data will en send to the DMA. This register is used for optical test usage since the Perceptron/N3 have only 10 to 8 bits LUT only. The DMA must be configured in synthesys to be able to transfert 10bpp images.	
Value at Reset:	0x0	
Possible Values:	0x0	Use LUT logic.
	0x1	LUT logic bypass.

LUT_PALETTE_USE	LUT PALETTE to USE	
RW	This register selects the LUT palette to be use in the grab path.	
Value at Reset:	0x0	
Possible Values:	0x0	Palette 0 is used
	0x1	Palette 1 is used

LUT_PALETTE_W	LUT PALETTE to Write	
RW	This register selects the palette to be write into the LUT. This register must be set to 0 when programming the Palette 0 and to 1 when programming the Palette 1.	
Value at Reset:	0x0	
Possible Values:	0x0	Write Palette 0
	0x1	Write Palette 1

LUT_DATA_W (9:0)	LUT DATA to Write
RW	Data to write in the LUT
Value at Reset:	0x0

LUT_SEL (2:0)	LUT SELection	
RW	LUT programmation selector. The Color and Mono shares the same 4 physical LUT.	
Value at Reset:	0x0	
Possible Values:	0x0	Read or Write to Gamma / Mono0 LUT
	0x1	Read or write to Blue / Mono1 LUT
	0x2	Read or write to Green / Mono2 LUT
	0x3	Read or write to Red / Mono3 LUT
	0x4	Write ALL LUT with same data.
	0x5	
	0x6	
	0x7	

LUT_WRN	LUT Write ReadNot	
RW	LUT Write mode	
Value at Reset:	0x0	
Possible Values:	0x0	Read operation
	0x1	Write operation

LUT_SS	LUT SnapShot
WO/AutoClr	Start the LUT OPERATION (R/W)

LUT_ADD (9:0)	
RW	
Value at Reset:	0x0

LUT_RB

Address: section "DATA" base address + 0x008

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

LUT_RB (9:0)	
RO	LUT ReadBack

WB_MULT1

Address: section "DATA" base address + 0x010

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

WB_MULT_G (15:0) *RW*	
Value at Reset:	0x1000

WB_MULT_B (15:0) *RW*	
Value at Reset:	0x1000

WB_MULT2

Address: section "DATA" base address + 0x018

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

WB_MULT_R (15:0) *RW*	
Value at Reset:	0x1000

WB_B_ACC

Address: section "DATA" base address + 0x020

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

B_ACC (30:0)	
RO	ACQuisition Blue ACCumulator

WB_G_ACC

Address: section "DATA" base address + 0x028

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

G_ACC (31:0)	
RO	ACQuisition Green ACCumulator

WB_R_ACC

Address: section "DATA" base address + 0x030

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

R_ACC (30:0)	
RO	ACQuisition Red ACCumulator

FPN_ADD

Address: section "DATA" base address + 0x038

31	30	29	28	27	26	25	24
FPN_73	Reserved		FPN_WE	Reserved			FPN_EN
23	22	21	20	19	18	17	16
Reserved							FPN_SS
15	14	13	12	11	10	9	8
Reserved						FPN_ADD(9:8)	
7	6	5	4	3	2	1	0
FPN_ADD(7:0)							

FPN_73		
RW	Use [7].[3] fpn correction instead old [5].[3] . This 7.3 mode is not implemented in the released FPGA.	
Value at Reset:	0x0	
Possible Values:	0x0	Use normal fpn mode 5.3
	0x1	Use advanced fpn mode 7.3

FPN_WE	FPN Write Enable	
RW	This register is the coefficient RAM WRITE ENABLE	
Value at Reset:	0x1	
Possible Values:	0x0	Read operation
	0x1	Write operation

FPN_EN	FPN ENable	
RW	This field enables the HW FPN and PRNU correction	
Value at Reset:	0x0	
Possible Values:	0x0	HW correction disable
	0x1	HW correction enable

FPN_SS	FPN SnapShot	
WO/AutoClr	This register is the snapshot for read/write to the coefficient RAM.	
Possible Values:	0x0	Nothing
	0x1	Snapshot

FPN_ADD (9:0)	FPN ADDress
RW	This register is the address to be write/read in the coefficient RAM. The first 512(144bits) locations are correction factors to not SUBsampled image(palette 0). The second 512 locations(144bits) are correction factors to SUBsampled image(palette 1).
Value at Reset:	0x0

FPN_READ_REG

Address: section "DATA" base address + 0x03C

31	30	29	28	27	26	25	24
Reserved	FPN_READ_PIX_SEL(2:0)			Reserved			FPN_READ_PRNU(8)
23	22	21	20	19	18	17	16
FPN_READ_PRNU(7:0)							
15	14	13	12	11	10	9	8
Reserved					FPN_READ_FPN(10:8)		
7	6	5	4	3	2	1	0
FPN_READ_FPN(7:0)							

FPN_READ_PIX_SEL (2:0)		
RW	This is the pixel number to be read (0 to 7) in the RAM. Each RAM location contains corrections for 8 pixels per address(FPN_ADD). This field selects the PIXel correction to be readed.	
Value at Reset:	0x0	
Possible Values:	0x0 - 0x7	

FPN_READ_PRNU (8:0)	
RO	This is the PRNU coefficient readed in RAM.

FPN_READ_FPN (10:0)	
RO	This is the FPN coefficient readed in RAM.

FPN_DATA (7:0)

Address: section "DATA" base address + 0x040 + (index * 0x4)

31	30	29	28	27	26	25	24
Reserved							FPN_DATA_ PRNU(8)
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
FPN_DATA_PRNU(7:0)							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					FPN_DATA_FPN(10:8)		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FPN_DATA_FPN(7:0)							

FPN_DATA_PRNU (8:0)	FPN DATA PRNU

RW	This is the PRNU coefficient be written in RAM.
	
	PRNU factor is signed 9 bits [0].[00][+/-][8]
	
	From the DoubleValue calculated in SW, program this field as:
	
	FPN_DATA_PRNU = int(DoubleVal*2048.0)
	
	Clip correction to implement in the driver is :
	
	if(DoubleVal > (255.0 / 2048.0)) DoubleVal = (255.0 / 2048.0) (0.124511718)
	if(DoubleVal < -(255.0 / 2048.0)) DoubleVal = -(255.0 / 2048.0) (0.124511718)

Value at Reset:	0x0

FPN_DATA_FPN (10:0)	FPN DATA FPN

RW	This is the FPN coefficient be written in RAM.
	
	If FPN 5.3 is implemented(default) factor is signed 9 bits [+/-][5].[3]
	If FPN 7.3 is implemented(default) factor is signed 11 bits [+/-][7].[3]
	
	In 5.3 configuration, from the DoubleValue alculated in SW, program this field as:
	
	FPN_DATA_FPN = int(DoubleVal*8.0)
	
	Clip correction to implement in the driver is :
	
	if(DoubleValue > 255.0/8.0) DoubleValue= 255.0/8.0 (31.875)
	if(DoubleValue < -255.0/8.0) DoubleValue= -255.0/8.0 (-31.875)

Value at Reset:	0x0

FPN_CONTRAST

Address: section "DATA" base address + 0x060

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

CONTRAST_GAIN (11:0)	
RW	This is a digital gain [4].[8] applied after the EXIT_CP3 substractor. This register MUST be set to 1 or greater.
Value at Reset:	0x100

CONTRAST_OFFSET (7:0)	CONTRAST OFFSET
RW	This is the constant substracted to the 10 bit pixel FPN and PRNU corrected. The value is a 8 bits integer value [8].[0] . This register is aligned with the LSB of the 10 bit pixel value.
Value at Reset:	0x0

FPN_ACC_ADD

Address: section "DATA" base address + 0x068

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		FPN_ACC_MODE_SEL	FPN_ACC_MODE_EN	Reserved			FPN_ACC_R_SS
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				FPN_ACC_ADD(11:8)			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FPN_ACC_ADD(7:0)							

FPN_ACC_MODE_SEL		
RW	This register selects if the Contrast Gain and Offset is used for compute the pixel accumulators. If FPN_ACC_MODE_SEL =0 then the module will use CONTRAST_GAIN=1 and CONTRAST_OFFSET=0 for the accumulators. If FPN_ACC_MODE_SEL =1 then the module will use Gain and Offset from registers CONTRAST_GAIN and CONTRAST_OFFSET.	
Value at Reset:	0x0	
Possible Values:	0x0	Don't use Contrast Gain and Offset
	0x1	Use Contrast Gain and Offset

FPN_ACC_MODE_EN	FPN ACCumulator MODE ENable	
RW	This field defines the accumulator mode. When this register is set to '1', the accumulators will start count and no frame will be sent to the host memory.	
Value at Reset:	0x0	
Possible Values:	0x0	Normal DMA transfert mode
	0x1	Accumulator mode

FPN_ACC_R_SS	FPN ACCumulator Read Snapshot
WO/AutoClr	This is the column read accumulator snapshot.

FPN_ACC_ADD (11:0)	FPN ACCumulator ADDress
RW	This is the column accumulator adress to read.
Value at Reset:	0x0

FPN_ACC_DATA

Address: section "DATA" base address + 0x070

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

FPN_ACC_R_WORKING	FPN ACCumulator Read WORKING	
RO	This field is the working status of the read-to-column accumulator. The data in the field FPN_ACC_DATA will be valid when FPN_ACC_R_WORKING is set to '0'	
Possible Values:	0x0	The data in the field FPN_ACC_DATA is valid
	0x1	The data in the field FPN_ACC_DATA is invalid

FPN_ACC_DATA (23:0)	FPN ACCumulator DATA	
RO	This is the column accumulator.	
Possible Values:		Any 24 bits value

DPC_LIST_CTRL

Address: section "DATA" base address + 0x080

31	30	29	28	27	26	25	24
dpc_fifo_underrun	dpc_fifo_overrun	Reserved	dpc_fifo_reset	Reserved	dpc_firstlast_line_rem	dpc_pattern0_cfg	dpc_enable
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		dpc_list_count(5:0)					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			dpc_list_WRn	Reserved			dpc_list_ss
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		dpc_list_add(5:0)					

dpc_fifo_underrun		
RO	This is the fifo underrun status of the 2 linebuffers in the dpc macro. Write '1' then '0' to field dpc_FIFO_RST to reset this flag and reset the Fifo logic.	
Possible Values:	0x0	Underrun not detected
	0x1	Underrun detected

dpc_fifo_overrun		
RO	This is the fifo overrun status of the 2 linebuffers in the dpc macro. Write '1' then '0' to field dpc_FIFO_RST to reset this flag and reset the Fifo logic.	
Possible Values:	0x0	Overrun not detected
	0x1	Overrun detected

dpc_fifo_reset		
RW	Write '1' then '0' to field dpcL_FIFO_RST to reset overrun/underrun flags of the line buffers and reset the Fifo logic. The DPC dual port ram is not SW reset . The fifo in each processing DPC unit is HW reset at each SOF.	
Value at Reset:	0x0	
Possible Values:	0x0	Fifo in normal operation
	0x1	Fifo in reset State

dpc_firstlast_line_rem		
RW	When this register is set to 1, the DPC macro will remove the first and last line of the image corrected. This can be usefull if we want to correct the 4 pixels in the corners of the image. The SW can program two more lines in the frame so the DPC macro can have enough pixels to correct the 4 pixel coners.	
Value at Reset:	0x1	
Possible Values:	0x0	Do not remove any lines of the image received
	0x1	Remove first and last line of the image received

dpc_pattern0_cfg		
RW	This field configures the behabieur of the correction pattern 0x0. If this field is set to 0x0 then the current pixel will not be corrected. If this field is set to 0x1 then the current pixel will be replaced by the value 0x3ff (white pixel)	
Value at Reset:	0x1	
Possible Values:	0x0	Do not correct current pixel
	0x1	Replace current pixel by a white pixel (0x3ff)

dpc_enable		
RW	Dead Pixel Correction core Enable, when this field is set to 1, the DPC logic will correct all the dead pixels that are listed in the DPC list. The grab must be idle when changing this register.	
Value at Reset:	0x0	
Possible Values:	0x0	DPC logic is bypassed
	0x1	PDC logic is enable

dpc_list_count (5:0)		
RW	This is the number of entries in the DPC list. The driver need to set the dcp_list_count in order to correct the image. Up to 63 pixels can be corrected. The value 0 is allowed and when set to 0 no pixel will be corrected.	
Value at Reset:	0x0	
Possible Values:	Any Value	Any value from 0 to 63

dpc_list_WRn		
RW	This is the Write/ReadN flag. To write to the DPC list set this bit to 1 and start the transaction with the dpc_list_ss field. To read from the DPC list set this bit to 0 and start the transaction with the dpc_list_ss field.	
Value at Reset:	0x0	
Possible Values:	0x0	Read list operation
	0x1	Write list operation

dpc_list_ss		
WO/AutoClr	This is the DPC snapshot. In order to start a write or read transaction the snapsot needs to be writen to '1'. This bit is a auto clear regsiter.	
Possible Values:	0x0	Do nothing
	0x1	Start the READ/WRITE transaction

dpc_list_add (5:0)		
RW	This is the address of the DPC list to be access by the read/write operation. Pixel 0 to correct is located at address b000000. Since the dpc_list_count field is also 6 bit wide, address 0 to 62 of the list can be used. Adress 0x3f cannot be used. This DPC location will not be corrected.	
Value at Reset:	0x0	
Possible Values:	0x0 - 0x3E	Valid DPC adress

DPC_LIST_DATA

Address: section "DATA" base address + 0x084

31	30	29	28	27	26	25	24
dpc_list_corr_pattern(7:0)							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
dpc_list_corr_y(11:4)							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
dpc_list_corr_y(3:0)				dpc_list_corr_x(11:8)			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
dpc_list_corr_x(7:0)							

dpc_list_corr_pattern (7:0)	
RW	This is pattern of the pixel to be corrected when executing a write to the DPC list. 2 bit correction : 34, 17, 136, 68 4 bit correction : 170, 153, 51, 204, 85, 102 6 bit correcttion: 187,238 (mapped to 170), 119,221 (mapped to 85) 8 bit correction : 255 Set pixel to 255 (white), debug : 0
Value at Reset:	0x0

dpc_list_corr_y (11:0)	
RW	This is Y location of the pixel to be corrected when executing a write to the DPC list.
Value at Reset:	0x0

dpc_list_corr_x (11:0)	
RW	This is X location of the pixel to be corrected when executing a write to the DPC list.
Value at Reset:	0x0

DPC_LIST_DATA_RD

Address: section "DATA" base address + 0x088

31	30	29	28	27	26	25	24
dpc_list_corr_pattern(7:0)							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
dpc_list_corr_y(11:4)							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
dpc_list_corr_y(3:0)				dpc_list_corr_x(11:8)			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
dpc_list_corr_x(7:0)							

dpc_list_corr_pattern (7:0)	
RO	This is pattern of the pixel read from DPC list. 2 Bit correction : 34, 17, 136, 68 4 Bit correction : 170, 153, 51, 204, 85, 102 6 bit correcttion: 187,238 (mapped to 170), 119,221 (mapped to 85) 8 bit correction : 255 Set pixel to 255 (white), debug : 0

dpc_list_corr_y (11:0)	
RO	This is Y location of the pixel read from the DPC list.

dpc_list_corr_x (11:0)	
RO	This is X location of the pixel read from the DPC list.

Section: HISPI

Address Range: [0x400 - 0x404]

CTRL

Address: section "HISPI" base address + 0x000

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

CLR	
RW	
Value at Reset:	0x0

RESET_IDELAYCTRL	Reset the xilinx macro IDELAYCTRL
RW	
Value at Reset:	0x0

STATUS

Address: section "HISPI" base address + 0x004

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

PLL_LOCKED	
RO	