

Register file structure : regfile\_xgs\_ctrl.pdf

Created by jmansill on 2020/04/01 08:20:09

Register file CRC32 : 0xAB304108

## 1. Main Parameters

Register file endianness: little endian

Address bus width: 12 bits

Data bus width: 32 bits

## 2. Memory Map

Section name	Address(es) / Address Ranges	Register name	Access Type
SYSTEM	0x000	ID	R
	0x030	ACQ_CAP	R
ACQ	0x100	GRAB_CTRL	RW
	0x108	GRAB_STAT	R
	0x110	READOUT_CFG1	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_ADDDATA	RW
	0x168	ACQ_SER_STAT	R
	0x190	SENSOR_CTRL	RW
	0x198	SENSOR_STAT	R
	0x1A0	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
	0x1D8	SENSOR_M_LINES	RW
	0x1DC	SENSOR_F_LINES	RW
	0x1E0	DEBUG_PINS	RW
	0x1E8	TRIGGER_MISSED	RW

Section name	Address(es) / Address Ranges	Register name	Access Type
	0x1F0	<a href="#">SENSOR_FPS</a>	R
	0x2A0	<a href="#">DEBUG</a>	RW
	0x2A8	<a href="#">DEBUG_CNTR1</a>	R
	0x2B0	<a href="#">DEBUG_CNTR2</a>	R
	0x2B4	<a href="#">DEBUG_CNTR3</a>	R
	0x2B8	<a href="#">EXP_FOT</a>	RW
	0x2C0	<a href="#">ACQ_SFNC</a>	RW

Section: SYSTEM

Address Range: [0x000 - 0x030]

ID

Address: section "SYSTEM" base address + 0x000

Description:

Static ID

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

StaticID (31:0)	MINUTEs of the build
RO	

## ACQ\_CAP

Address: section "SYSTEM" base address + 0x030

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
DPC	EXP_FOT	FPN_73	COLOR	CH_LVDS(3:0)			
7	6	5	4	3	2	1	0
Reserved			LUT_WIDTH	Reserved		LUT_PALETTE(1:0)	

<b>DPC</b>		
<i>STATIC</i>	This field indicate if the Dead Pixel Corection is implemented in the fpga.	
Value at Reset:	0x1	
Possible Values:	0x0	DPC is not implemented
	0x1	DPC is implemented

<b>EXP_FOT</b>		
<i>STATIC</i>	This field indicate if the Exposure during FOT is implemented in the fpga.	
Value at Reset:	0x1	

<b>FPN_73</b>	FPN 7.3 correction CAP	
<i>RO</i>	This field indicate if the FPN correction 7.3 is implemented instead the default 5.3 in the FPGA acquisition.	
Possible Values:	0x0	Default 5.3 FPN correction implemented
	0x1	New 7.3 FPN correction implemented

<b>COLOR</b>		
<i>RO</i>	This field indicate if the COLOR path is implemented in the FPGA acquisition.	

<b>CH_LVDS (3:0)</b>		
<i>RO</i>	This is the number of LVDS DATA channels connected between the CMOS sensor and the FPGA. This number does not include the LVDS CTROL channel.	

<b>LUT_WIDTH</b>		
<i>RO</i>	This is the LUT width implemented in the acquisition core.	
Possible Values:	0x0	LUT 10 to 10 Bits
	0x1	LUT 10 to 8 Bits

<b>LUT_PALETTE (1:0)</b>		
<i>RO</i>	This is the number of LUT palette implemented in the acquisition module	
Possible Values:	0x0	No LUT implemented
	0x1	1 LUT implemented
	0x2	2 LUT implemented

## 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_O VERLAP_BUFn
15	14	13	12	11	10	9	8
TRIGGER_O VERLAP	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

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

<b>GRAB_ROI2_EN</b>	
<i>RW</i>	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

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

<b>TRIGGER_OVERLAP_BUF</b> <b>Fn</b> <i>RW</i>		
	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.

<b>TRIGGER_OVERLAP</b> <i>RW</i>		
	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)

<b>TRIGGER_ACT (2:0)</b> <i>RW</i>	<b>TRIGGER ACTivation</b>	
	<p>This is the trigger activation . This register selects the activation of the trigger when the trigger source is set to Hardware Snapshot mode . This register is Double Buffered, so the trigger activation may change from one grab command to another.</p> <p>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.</p> <p>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.</p>	
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

<b>TRIGGER_SRC (2:0)</b> <i>RW</i>	<b>TRIGGER SouRCe</b>	
	<p>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.</p>	
Value at Reset:	0x0	
Possible Values:	0x0	RESERVED
	0x1	Immediate mode (Continuous)
	0x2	Hardware Snapshot mode
	0x3	Software Snapshot mode
	0x4	SFNC mode (auto trig)

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

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

<b>GRAB_CMD</b> <i>WO/AutoClr</i>	GRAB CoMmanD	
	<p>This is MIL GRAB command.</p> <p>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 Snapshot mode or Hardware Snapshot mode, GRAB_CMD will act as an ARM.</p> <p>The GRAB_CMD will take around 13 clks to record the grab parameters to the SPI fifo. The GRAB_CMD_DONE register may be read to avoid fifo corruption before sending another Grab command instruction.</p>	
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_DELA I	ABORT_DONE E	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_READ OUT	GRAB_EXPO SURE	Reserved	GRAB_PEND ING	GRAB_ACTI VE	GRAB_IDLE

<b>GRAB_CMD_DONE</b> <i>RO</i>	GRAB CoMmanD DONE						
	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				

<b>ABORT_PET</b> <i>RO</i>	ABORT during PET						
	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				

<b>ABORT_DELA</b> <i>RO</i>							
	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				

<b>ABORT_DONE</b> <i>RO</i>	ABORT is DONE						
	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)				

<b>TRIGGER_RDY</b> <i>RO</i>							

<b>ABORT_MNGR_STAT (2:0)</b> <i>RO</i>							
	DEBUG ABORT MANAGER STATE MACHINE						

<b>TRIG_MNGR_STAT (3:0)</b>	
<i>RO</i>	DEBUG TRIGGER MANAGER STATE MACHINE

<b>TIMER_MNGR_STAT (2:0)</b>	
<i>RO</i>	DEBUG TIMER MANAGER STATE MACHINE

<b>GRAB_MNGR_STAT (3:0)</b>	
<i>RO</i>	DEBUG GRAB MANAGER STATE MACHINE

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

<b>GRAB_READOUT</b>	
<i>RO</i>	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.

<b>GRAB_EXPOSURE</b>		
<i>RO</i>	This is the sensor integration status	
Possible Values:	0x0	Idle
	0x1	Integrating

<b>GRAB_PENDING</b> <i>RO</i>		
	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

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

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

## READOUT\_CFG1

Address: section "ACQ" base address + 0x010

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

<b>GRAB_REVX_OVER_RST</b>	
<i>WO/AutoClr</i>	This field reset the reverseX overrun flag GRAB_REVX_OVER.

<b>GRAB_REVX_OVER</b>					
<i>RO</i>	This field informs the software that a reverseX overrun has been detected. When this status bit is set to '1' by the fpga, it will remain at '1' until it is reset by the field GRAB_REVX_OVER_RST.				
Possible Values:	<table> <tr> <td>0x0</td><td>No ReverseX Overrun detected</td></tr> <tr> <td>0x1</td><td>At least one ReverseX Overrun was detected</td></tr> </table>	0x0	No ReverseX Overrun detected	0x1	At least one ReverseX Overrun was detected
0x0	No ReverseX Overrun detected				
0x1	At least one ReverseX Overrun was detected				

<b>GRAB_REVX</b>	
<i>RW</i>	
Value at Reset:	0x0

<b>ROT_LENGTH (9:0)</b>	Row Overhead Time LENGTH
<i>STATIC</i>	This is the length of the Row Overhead Time. [NOT USED FOR THE MOMENT- FOR FUTURE USE]
Value at Reset:	0x0
Possible Values:	Any Value      Any 8 bits value

<b>FOT_LENGTH (15:0)</b>	Frame Overhead Time LENGTH
<i>STATIC</i>	This is the length of the Frame Overhead Time. [NOT USED FOR THE MOMENT- FOR FUTURE USE]
Value at Reset:	0x0
Possible Values:	Any Value      Any 16 bit value

## 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)							

<b>READOUT_LENGTH (28:0)</b>		
<i>RO</i>	This is the readout length register. This register is a register software calculated that gives the readout length to the grab engine. This register will depend on the ROI, Subsampling, Binning and LVDS channels used. It is used in the PET engine 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)							

<b>KEEP_OUT_TRIG_ENA</b>	
<i>RW</i>	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

<b>LINE_TIME (15:0)</b>	LINE TIME	
<i>RW</i>	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)							

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

<b>KEEP_OUT_TRIG_START (15:0)</b>	
<i>RW</i>	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)							

<b>EXPOSURE_LEV_MODE</b>	EXPOSURE LEVel MODE	
<i>RW</i>	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

<b>EXPOSURE_SS (27:0)</b>	EXPOSURE Single Slope	
<i>RW</i>	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)							

<b>EXPOSURE_DS (27:0)</b>	<b>EXPOSURE Dual</b>	
<i>RW</i>	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)							

<b>EXPOSURE_TS (27:0)</b> <i>RW</i>	<b>EXPOSURE Tripple</b>	
	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)							

<b>TRIGGER_DELAY (27:0)</b> <i>RW</i>	<b>TRIGGER_DELAY</b>	
	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_PO L	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)							

<b>STROBE_E</b> <i>RW</i>	<b>STROBE Enable</b>						
	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 programming.						
	This register is double buffered						
Value at Reset:	0x0						
Possible Values:	0x0	Strobe disabled					
	0x1	Strobe enabled					

<b>STROBE_POL</b> <i>RW</i>	<b>STROBE POLarity</b>						
	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					

<b>STROBE_START (27:0)</b> <i>RW</i>	<b>STROBE START</b>						
	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_MO DE	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)							

<b>STROBE_MODE</b> <i>RW</i>	<b>STROBE MODE</b>	
	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

<b>STROBE_B_EN</b> <i>RW</i>	<b>STROBE phase B ENable</b>	
	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

<b>STROBE_A_EN</b> <i>RW</i>	<b>STROBE phase A ENable</b>	
	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

<b>STROBE_END (27:0)</b> <i>RW</i>	<b>STROBE END</b>	
	This is the strobe end location. This location does not depend on the Strobe Mode used.	
	This register is double buffered	
Value at Reset:	0xffffffff	
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

<b>SER_RWn</b>	SERial Read/Writen	
<i>RW</i>	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

<b>SER_CMD (1:0)</b>	SERial CoMmand	
<i>RW</i>	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

<b>SER_RF_SS</b>	SERial Read Fifo SnapShot	
<i>WO/AutoClr</i>	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

<b>SER_WF_SS</b>	SERial Write Fifo SnapShot	
<i>WO/AutoClr</i>	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)							

<b>SER_DAT (15:0)</b>	SERial interface DATA	
<i>RW</i>	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

<b>SER_ADD (14:0)</b>	SERial interface ADDRESS	
<i>RW</i>	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)							

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

<b>SER_BUSY</b>	SERial BUSY	
<i>RO</i>	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 running

<b>SER_DAT_R (15:0)</b>	SERial interface DATa Read	
<i>RO</i>	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_UPDATE	Reserved		SENSOR_RESETN	SENSOR_POWERUP

<b>SENSOR_REFRESH_TEMP</b> <i>WO/AutoClr</i>	SENSOR REFRESH TEMPerature						
	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'.						
Possible Values:	0x0	Idle					
	0x1	Starts a Temperature read on Python SPI interface					

<b>SENSOR_POWERDOWN</b> <i>WO/AutoClr</i>							
	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.						

<b>SENSOR_COLOR</b> <i>RW</i>	SENSOR COLOR						
	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	Monochrome sensor					
	0x1	Color sensor					

<b>SENSOR_REG_UPDATE</b> <i>RW</i>	SENSOR REGister UPDATE						
	By setting this bit to 1, the SENSOR CONTROLLER WILL UPDATE the prograded CMOS sensor registers at the beginning of each grab.						
Value at Reset:	0x1						
Possible Values:	0x0	Do not update registers					
	0x1	Update registers					

<b>SENSOR_RESETN</b> <i>RW</i>	SENSOR RESET Not						
	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					

<b>SENSOR_POWERUP</b>		
<i>WO/AutoClr</i>	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_STATUS	SENSOR_POWERUP_DONE

<b>SENSOR_TEMP (7:0)</b>		
<i>RO</i>	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.	
Possible Values:	Any Value	

<b>SENSOR_TEMP_VALID</b>	SENSOR TEMPerature VALID	
<i>RO</i>	This field indicates that the field SENSOR_TEMP have valid temperature after a SENSOR_REFRESH_TEMP snapshot.	
Possible Values:	0x0	SENSOR_TEMPERATURE register is not valid
	0x1	SENSOR_TEMPERATURE register is valid

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

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

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

<b>SENSOR_VCC_PG</b>	SENSOR supply VCC Power Good	
<i>RO</i>	This is the VCC Power Good status (generated by external HW).	
Possible Values:	0x0	Disable
	0x1	Enable

<b>SENSOR_POWERUP_STAT</b> <i>RO</i>		
	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

<b>SENSOR_POWERUP_DONE</b> <i>RO</i>		
	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 + 0x0A0

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

<b>reserved1 (11:0)</b>	
<i>STATIC</i>	
Value at Reset:	0x0

<b>ACTIVE_SUBSAMPLING_Y</b>	
<i>RW</i>	Subsampling (Row) for ROI Configurations
Value at Reset:	0x0
Possible Values:	0x0
	0x1

<b>reserved0</b>	
<i>STATIC</i>	
Value at Reset:	0x0
Possible Values:	0x0
	Idle
	0x1
	Enable

<b>M_SUBSAMPLING_Y</b>	
<i>RW</i>	Subsampling (Row) for M Region
Value at Reset:	0x0
Possible Values:	0x0
	0x1

<b>SUBSAMPLING_X</b>	
<i>RW</i>	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)							

<b>reserved1 (4:0)</b>	
<i>STATIC</i>	
Value at Reset:	0x0

<b>ANALOG_GAIN (2:0)</b>	
<i>RW</i>	
Value at Reset:	0x1
Possible Values:	0x1      1x
	0x3      2x
	0x7      4x

<b>reserved0 (7:0)</b>	
<i>STATIC</i>	
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)							

<b>reserved (5:0)</b>	
<i>STATIC</i>	
Value at Reset:	0x0

<b>Y_START (9:0)</b>	Y START
<i>RW</i>	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)							

<b>reserved (5:0)</b>	
<i>STATIC</i>	
Value at Reset:	0x0

<b>Y_START (9:0)</b>	Y START
<i>RW</i>	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 + 0x0D8

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)							

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

<b>M_LINES_SENSOR (9:0)</b>	
<i>RW</i>	Number of Lines to Readout from M Region in Context 0 Unit is #lines Total number of Black lines = M_LINES-M_SUPPRESSED
Value at Reset:	0x8

## SENSOR F\_LINES

Address: section "ACQ" base address + 0x0DC

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	F_SUPPRESSED(4:0)					F_LINES_SENSOR(9:8)	
7	6	5	4	3	2	1	0
F_LINES_SENSOR(7:0)							

<b>F_SUPPRESSED (4:0)</b>	
<i>RW</i>	Suppress the Readout of Initial Lines in the F Region , Unit is #lines
Value at Reset:	0x0

<b>F_LINES_SENSOR (9:0)</b>	
<i>RW</i>	Number of Lines to Read from F Region in Context 0 Unit is #lines
Value at Reset:	0x8

## 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)			

<b>Debug3_sel (4:0)</b> <i>RW</i>	
	<pre> debug_vector(0x0) &lt;= python_monitor0; debug_vector(0x1) &lt;= python_monitor1; debug_vector(0x2) &lt;= grab_mgr_trig_rdy; debug_vector(0x3) &lt;= curr_trig0; debug_vector(0x4) &lt;= strobe; debug_vector(0x5) &lt;= python_exposure; debug_vector(0x6) &lt;= FOT; debug_vector(0x7) &lt;= readout; debug_vector(0x8) &lt;= readout_stateD; debug_vector(0x9) &lt;= ext_trig; debug_vector(0xa) &lt;= REGFILE.ACQ.GRAB_CTRL.GRAB_CMD; debug_vector(0xb) &lt;= REGFILE.ACQ.GRAB_CTRL.GRAB_SS; debug_vector(0xc) &lt;= grab_mgr_trig; debug_vector(0xd) &lt;= grab_mgr_trig_rdy; debug_vector(0xe) &lt;= grab_pending; debug_vector(0xf) &lt;= grab_active; debug_vector(0x10) &lt;= DEC_DATA_EN; debug_vector(0x11) &lt;= DEC_SOL; debug_vector(0x12) &lt;= DEC_SOF; debug_vector(0x13) &lt;= DEC_EOL; debug_vector(0x14) &lt;= DEC_EOF; debug_vector(0x15) &lt;= DEC_CRC; debug_vector(0x16) &lt;= DEC_TRAIN; debug_vector(0x17) &lt;= fpnprnu_corr_sof; debug_vector(0x18) &lt;= fpnprnu_corr_sol; debug_vector(0x19) &lt;= fpnprnu_corr_data_val; debug_vector(0x1a) &lt;= fpnprnu_corr_eol; debug_vector(0x1b) &lt;= fpnprnu_corr_eof; debug_vector(0x1c) &lt;= python_ssn_int; debug_vector(0x1d) &lt;= debug_lvds(0); debug_vector(0x1e) &lt;= debug_lvds(1); debug_vector(0x1f) &lt;= 'Z'; </pre>
Value at Reset:	0x1f

<b>Debug2_sel (4:0)</b> <i>RW</i>	<pre> debug_vector(0x0) &lt;= python_monitor0; debug_vector(0x1) &lt;= python_monitor1; debug_vector(0x2) &lt;= grab_mgr_trig_rdy; debug_vector(0x3) &lt;= curr_trig0; debug_vector(0x4) &lt;= strobe; debug_vector(0x5) &lt;= python_exposure; debug_vector(0x6) &lt;= FOT; debug_vector(0x7) &lt;= readout; debug_vector(0x8) &lt;= readout_stateD; debug_vector(0x9) &lt;= ext_trig; debug_vector(0xa) &lt;= REGFILE.ACQ.GRAB_CTRL.GRAB_CMD; debug_vector(0xb) &lt;= REGFILE.ACQ.GRAB_CTRL.GRAB_SS; debug_vector(0xc) &lt;= grab_mgr_trig; debug_vector(0xd) &lt;= grab_mgr_trig_rdy; debug_vector(0xe) &lt;= grab_pending; debug_vector(0xf) &lt;= grab_active; debug_vector(0x10) &lt;= DEC_DATA_EN; debug_vector(0x11) &lt;= DEC_SOL; debug_vector(0x12) &lt;= DEC_SOF; debug_vector(0x13) &lt;= DEC_EOL; debug_vector(0x14) &lt;= DEC_EOF; debug_vector(0x15) &lt;= DEC_CRC; debug_vector(0x16) &lt;= DEC_TRAIN; debug_vector(0x17) &lt;= fpnprnu_corr_sof; debug_vector(0x18) &lt;= fpnprnu_corr_sol; debug_vector(0x19) &lt;= fpnprnu_corr_data_val; debug_vector(0x1a) &lt;= fpnprnu_corr_eol; debug_vector(0x1b) &lt;= fpnprnu_corr_eof; debug_vector(0x1c) &lt;= python_ssn_int; debug_vector(0x1d) &lt;= debug_lvds(0); debug_vector(0x1e) &lt;= debug_lvds(1); debug_vector(0x1f) &lt;= 'Z'; </pre>
Value at Reset:	0x1f

<b>Debug1_sel (4:0)</b> <i>RW</i>	<pre> debug_vector(0x0) &lt;= python_monitor0; debug_vector(0x1) &lt;= python_monitor1; debug_vector(0x2) &lt;= grab_mgr_trig_rdy; debug_vector(0x3) &lt;= curr_trig0; debug_vector(0x4) &lt;= strobe; debug_vector(0x5) &lt;= python_exposure; debug_vector(0x6) &lt;= FOT; debug_vector(0x7) &lt;= readout; debug_vector(0x8) &lt;= readout_stateD; debug_vector(0x9) &lt;= ext_trig; debug_vector(0xa) &lt;= REGFILE.ACQ.GRAB_CTRL.GRAB_CMD; debug_vector(0xb) &lt;= REGFILE.ACQ.GRAB_CTRL.GRAB_SS; debug_vector(0xc) &lt;= grab_mgr_trig; debug_vector(0xd) &lt;= grab_mgr_trig_rdy; debug_vector(0xe) &lt;= grab_pending; debug_vector(0xf) &lt;= grab_active; debug_vector(0x10) &lt;= DEC_DATA_EN; debug_vector(0x11) &lt;= DEC_SOL; debug_vector(0x12) &lt;= DEC_SOF; debug_vector(0x13) &lt;= DEC_EOL; debug_vector(0x14) &lt;= DEC_EOF; debug_vector(0x15) &lt;= DEC_CRC; debug_vector(0x16) &lt;= DEC_TRAIN; debug_vector(0x17) &lt;= fpnprnu_corr_sof; debug_vector(0x18) &lt;= fpnprnu_corr_sol; debug_vector(0x19) &lt;= fpnprnu_corr_data_val; debug_vector(0x1a) &lt;= fpnprnu_corr_eol; debug_vector(0x1b) &lt;= fpnprnu_corr_eof; debug_vector(0x1c) &lt;= python_ssn_int; debug_vector(0x1d) &lt;= debug_lvds(0); debug_vector(0x1e) &lt;= debug_lvds(1); debug_vector(0x1f) &lt;= 'Z'; </pre>
Value at Reset:	0x1f

<b>Debug0_sel (4:0)</b> RW	<pre> debug_vector(0x0) &lt;= python_monitor0; debug_vector(0x1) &lt;= python_monitor1; debug_vector(0x2) &lt;= grab_mngr_trig_rdy; debug_vector(0x3) &lt;= curr_trig0; debug_vector(0x4) &lt;= strobe; debug_vector(0x5) &lt;= python_exposure; debug_vector(0x6) &lt;= FOT; debug_vector(0x7) &lt;= readout; debug_vector(0x8) &lt;= readout_stateD; debug_vector(0x9) &lt;= ext_trig; debug_vector(0xa) &lt;= REGFILE.ACQ.GRAB_CTRL.GRAB_CMD; debug_vector(0xb) &lt;= REGFILE.ACQ.GRAB_CTRL.GRAB_SS; debug_vector(0xc) &lt;= grab_mngr_trig; debug_vector(0xd) &lt;= grab_mngr_trig_rdy; debug_vector(0xe) &lt;= grab_pending; debug_vector(0xf) &lt;= grab_active; debug_vector(0x10) &lt;= DEC_DATA_EN; debug_vector(0x11) &lt;= DEC_SOL; debug_vector(0x12) &lt;= DEC_SOF; debug_vector(0x13) &lt;= DEC_EOL; debug_vector(0x14) &lt;= DEC_EOF; debug_vector(0x15) &lt;= DEC_CRC; debug_vector(0x16) &lt;= DEC_TRAIN; debug_vector(0x17) &lt;= fpnprnu_corr_sof; debug_vector(0x18) &lt;= fpnprnu_corr_sol; debug_vector(0x19) &lt;= fpnprnu_corr_data_val; debug_vector(0x1a) &lt;= fpnprnu_corr_eol; debug_vector(0x1b) &lt;= fpnprnu_corr_eof; debug_vector(0x1c) &lt;= python_ssn_int; debug_vector(0x1d) &lt;= debug_lvds(0); debug_vector(0x1e) &lt;= debug_lvds(1); debug_vector(0x1f) &lt;= 'Z'; </pre>
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) RO	SENSOR Frame Per Second
	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		TEST_MODE_PIX_START(9:8)	
23	22	21	20	19	18	17	16
TEST_MODE_PIX_START(7:0)							
15	14	13	12	11	10	9	8
Reserved						TEST_MOVE	TEST_MODE
7	6	5	4	3	2	1	0
LED_STAT_CLHS(1:0)		LED_STAT_CTRL(1:0)		Reserved	LED_TEST_COLOR(1:0)		LED_TEST

<b>DEBUG_RST_CNTR</b>		
<i>RW</i>	This register clears the debug cntrs	
Value at Reset:	0x1	
Possible Values:	0x0	
	0x1	Reset counters

<b>TEST_MODE_PIX_START (9:0)</b>		
<i>RW</i>	This register defines the value of the first pixel in the frame when the TEST_MODE is activated. In 8 bits mode only 8-MSB bits of the register is used.	
Value at Reset:	0x0	

<b>TEST_MOVE</b>		
<i>RW</i>	This field when in TEST_MODE=1, makes the ramp move. The first pixel of the frame is incremented by one each frame.	
Value at Reset:	0x0	
Possible Values:	0x0	Static test ramp
	0x1	The test ramp moves

<b>TEST_MODE</b>		
<i>RW</i>	This field set the FPGA in test mode. The fpga will send a programmable ramp to the host using the syncs received from the sensor. The generated ramp can move when set with the field TEST_MOVE.  In color mode (LVDSx1), the ramp pixel is repeated 3 times to generate a B&W ramp in RGB24 mode.	
Value at Reset:	0x0	
Possible Values:	0x0	Normal acquisition data from sensor
	0x1	Test mode, a ramp is generated.

<b>LED_STAT_CLHS (1:0)</b>		
<i>RO</i>		

<b>LED_STAT_CTRL (1:0)</b> <i>RO</i>		

<b>LED_TEST_COLOR (1:0)</b> <i>RW</i>		
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

<b>LED_TEST</b> <i>RW</i>		
	This register will put the LED status in test mode. The test mode is controlled by <b>LED_TEST_COLOR</b>	
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
EOF_CNTR(31:24)							
23	22	21	20	19	18	17	16
EOF_CNTR(23:16)							
15	14	13	12	11	10	9	8
EOF_CNTR(15:8)							
7	6	5	4	3	2	1	0
EOF_CNTR(7:0)							

EOF_CNTR (31:0) RO	
	This is the EOF CNTR. This feature is enabled by setting register regfile.ACQ.DEBUG.DEBUG_RST_CNTR to 0.

**DEBUG\_CNTR2**

Address: section "ACQ" base address + 0x1B0

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				EOL_CNTR(11:8)			
7	6	5	4	3	2	1	0
EOL_CNTR(7:0)							

EOL_CNTR (11:0) RO	
	This is the EOL CNTR. This feature is enabled by setting register regfile.ACQ.DEBUG.DEBUG_RST_CNTR to 0.

**DEBUG\_CNTR3**

Address: section "ACQ" base address + 0x1B4

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)							

<b>EXP_FOT</b> <i>RW</i>	EXPosure during FOT	
	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

<b>EXP_FOT_TIME (11:0)</b> <i>RW</i>	EXPosure during FOT TIME	
	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

<b>RELOAD_GRAB_PARAMS</b>		
<i>RW</i>	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	