

Register file structure : regfile\_ares.pdf

Created by amarchan on 2020/09/21 16:48:33

Register file CRC32 : 0xF5A7196A

## 1. Main Parameters

Register file endianness: little endian

Address bus width: 15 bits

Data bus width: 32 bits

## 2. Memory Map

 Note: A grey filled row indicates that the register read back capability is disabled for this register.

Section name	Address(es) / Address Ranges	Register name	Access Type
Device_specific	0x0000	INTSTAT	RW
	0x0004	INTMASKn	RW
	0x0008	INTSTAT2	RW
	0x001C	BUILDID	R
	0x0020	FPGA_ID	RW
	0x0024	LED_OVERRIDE	RW
INTERRUPT_QUEUE	0x0040	CONTROL	RW
	0x0044	CONS_IDX	RW
	0x0048	ADDR_LOW	RW
	0x004C	ADDR_HIGH	RW
	0x0050	MAPPING	W
SPI	0x00E0	SPIREGIN	RW
	0x00E8	SPIREGOUT	R
arbiter	0x00F0	ARBITER_CAPABILITIES	R
	0x00F4, 0x00F8	AGENT (1:0)	RW
IO [0]	0x0200	CAPABILITIES_IO	R
	0x0204	IO_PIN	R
	0x0208	IO_OUT	RW
	0x020C	IO_DIR	RW
	0x0210	IO_POL	RW
	0x0214	IO_INTSTAT	RW
	0x0218	IO_INTMASKn	RW
	0x021C	IO_ANYEDGE	RW
IO [1]	[0x0280 - 0x02FC]	...	...
Quadrature [0]	0x0300	CAPABILITIES_QUAD	R
	0x0304	PositionReset	RW
	0x0308	DecoderInput	RW
	0x030C	DecoderCfg	RW

Section name	Address(es) / Address Ranges	Register name	Access Type
	0x0310	<a href="#">DecoderPosTrigger</a>	RW
	0x0314	<a href="#">DecoderCntrLatch_Cfg</a>	RW
	0x0334	<a href="#">DecoderCntrLatched_SW</a>	R
	0x0338	<a href="#">DecoderCntrLatched</a>	R
<a href="#">TickTable [0]</a>	0x0380	<a href="#">CAPABILITIES_TICK_TBL</a>	R
	0x0384	<a href="#">CAPABILITIES_EXT1</a>	R
	0x0388	<a href="#">TickTableClockPeriod</a>	R
	0x038C	<a href="#">TickConfig</a>	RW
	0x0390	<a href="#">CurrentStampLatched</a>	R
	0x0394	<a href="#">WriteTime</a>	RW
	0x0398	<a href="#">WriteCommand</a>	RW
	0x039C	<a href="#">LatchIntStat</a>	RW
	0x03A0, 0x03A4	<a href="#">InputStamp (1:0)</a>	RW
	0x03A8, 0x03AC, ... ,0x03CC	<a href="#">reserved_for_extra_latch (9:0)</a>	R
	0x03D0, 0x03D4	<a href="#">InputStampLatched (1:0)</a>	R
<a href="#">InputConditioning</a>	0x0400	<a href="#">CAPABILITIES_INCOND</a>	R
	0x0404, 0x0408, ... ,0x0410	<a href="#">InputConditioning (3:0)</a>	RW
<a href="#">OutputConditioning</a>	0x0480	<a href="#">CAPABILITIES_OUTCOND</a>	R
	0x0484, 0x0488, ... ,0x0490	<a href="#">OutputCond (3:0)</a>	RW
	0x0494	<a href="#">Reserved</a>	R
	0x04AC	<a href="#">Output_Debounce</a>	RW
<a href="#">InternalInput</a>	0x0500	<a href="#">CAPABILITIES_INT_INP</a>	R
<a href="#">InternalOutput</a>	0x0580	<a href="#">CAPABILITIES_INTOUT</a>	R
	0x0584	<a href="#">OutputCond (0:0)</a>	RW
<a href="#">Timer [0]</a>	0x0600	<a href="#">CAPABILITIES_TIMER</a>	R
	0x0604	<a href="#">TimerClockPeriod</a>	R
	0x0608	<a href="#">TimerTriggerArm</a>	RW
	0x060C	<a href="#">TimerClockSource</a>	RW
	0x0610	<a href="#">TimerDelayValue</a>	RW
	0x0614	<a href="#">TimerDuration</a>	RW
	0x0618	<a href="#">TimerLatchedValue</a>	R
	0x061C	<a href="#">TimerStatus</a>	RW
<a href="#">Timer [1]</a>	[0x0680 - 0x06FC]	...	...
<a href="#">Timer [2]</a>	[0x0700 - 0x077C]	...	...
<a href="#">Timer [3]</a>	[0x0780 - 0x07FC]	...	...
<a href="#">Timer [4]</a>	[0x0800 - 0x087C]	...	...
...		...	...
<a href="#">Timer [7]</a>	[0x0980 - 0x09FC]	...	...
<a href="#">Microblaze</a>	0x0A00	<a href="#">CAPABILITIES_MICRO</a>	R
	0x0A04, 0x0A08	<a href="#">ProdCons (1:0)</a>	R
<a href="#">AnalogOutput</a>	0x0A80	<a href="#">CAPABILITIES_ANALOUT</a>	R
	0x0A84	<a href="#">OutputValue</a>	RW
<a href="#">EOFM</a>	0x0B00	<a href="#">EOFM</a>	R
<a href="#">ProdCons [0]</a>	0x2000	<a href="#">Pointers</a>	RW
	0x3000, 0x3004, ..., 0x3FFC	<a href="#">DPRAM (1023:0)</a>	RW
<a href="#">ProdCons [1]</a>	[0x4000 - 0x5FFC]	...	...

### 3. Registers definition

## Section: Device\_specific

Address Range: [0x0000 - 0x0024]

### INTSTAT

Address: section "Device\_specific" base address + 0x0000

Description:

INTerrupt STATus

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
IRQ_TICK_L ATCH	IRQ_MICRO BLAZE	Reserved	IRQ_TICK_W A	IRQ_TIMER	Reserved	IRQ_TICK	IRQ_IO

<b>IRQ_TICK_LATCH</b> <i>RW2C</i>							
	This bit indicates that an interrupt has been detected on one of the latch associated with the matching tick table						
Value at Reset:	0x0						
Possible Values:	0x0	No interrupt detected					
	0x1	Interrupt event occurred in the tick table					

<b>IRQ_MICROBLAZE</b> <i>RW2C</i>							
	This bit indicates that the Microblaze subsystem has updated datapointer to exchange data with the host.						
Value at Reset:	0x0						
Possible Values:	0x0	No interrupt detected.					
	0x1	This bit indicates that an interrupt has been detected.					

<b>IRQ_TICK_WA</b> <i>RW2C</i>							
	This bit indicates that a wrap around of the tick table X has happen.						
Value at Reset:	0x0						
Possible Values:	0x0	No interrupt detected.					
	0x1	This bit indicates that an interrupt has been detected. A wrap around of the tick table has been generated.					

<b>IRQ_TIMER</b> <i>RO</i>		
	This bit indicates that an interrupt has been detected on one Timer, see INTSTAT2 register. This is a read-only register.	
Possible Values:	0x0	No interrupt detected
	0x1	Interrupt event occurred in one Timer

<b>IRQ_TICK</b> <i>RW2C</i>		
	This bit indicates that an interrupt has been detected on the Tick Table X(half/full).	
Value at Reset:	0x0	
Possible Values:	0x0	No interrupt detected
	0x1	Interrupt event occurred in the Tick Table

<b>IRQ_IO</b> <i>RW2C</i>		
	This bit indicates that an interrupt has been detected on the User Inputs.	
Value at Reset:	0x0	
Possible Values:	0x0	No interrupt detected
	0x1	Interrupt event occurred in the User Inputs

## INTMASKn

Address: section "Device\_specific" base address + 0x0004

Description:

Every bit in this register is used to mask some event. While a bit in this register is set to 0, the corresponding bit in the INTSTAT register cannot SWITCH to 1. When a bit in the INTMASK register is set to '1', the corresponding bit in INTSTAT function normally

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
IRQ_TICK_L ATCH	IRQ_MICRO BLAZE	Reserved	IRQ_TICK_W A	IRQ_TIMER	Reserved	IRQ_TICK	IRQ_IO

<b>IRQ_TICK_LATCH</b> <i>STATIC</i>							
	This bit enable the interrupt on the latch from the tick table.  Note that it is hardcoded to 1. Individual interrupt for every latch should be enabled within the tick table feature structure.						
Value at Reset:	0x1						
Possible Values:	0x0	No interrupt detected					
	0x1	Interrupt event occurred in the tick table					

<b>IRQ_MICROBLAZE</b> <i>RW</i>							
	This is the IRQ MASKn for the Microblaze datapointers . When set this field to '0', no interrupt will be generated from the Microblaze datapointers update.						
Value at Reset:	0x0						
Possible Values:	0x0	IRQ from Microblaze disabled					
	0x1	IRQ from Microblaze enabled					

<b>IRQ_TICK_WA</b> <i>RW</i>							
	This is the IRQ MASKn for the Tick Table X Wrap Around . When set this field to '0', no interrupt will be generated from the Tick Table X Wrap Around.						
Value at Reset:	0x0						
Possible Values:	0x0	IRQ from Tick Table Wrap Around disabled					
	0x1	IRQ from Tick Table Wrap Around enabled					

<b>IRQ_TIMER</b> <i>RW</i>							
	This is the IRQ MASKn for the Timer IRQ. When set this field to '0', no interrupt will be generated from Timer module.						
Value at Reset:	0x0						
Possible Values:	0x0	IRQ from Timers pins disabled					
	0x1	IRQ from Timers enabled					

<b>IRQ_TICK</b>		
<i>RW</i>	This is the IRQ MASKn for the Tick Table. When set this field to '0', no interrupt will be generated from the Tick Table X.	
Value at Reset:	0x0	
Possible Values:	0x0	IRQ from Tick Table disabled
	0x1	IRQ from Tick Table enabled

<b>IRQ_IO</b>		
<i>RW</i>	This is the IRQ MASKn for the UserInputs pins interrupt. When set this field to '0', no interrupt will be generated from the input pins.	
Value at Reset:	0x0	
Possible Values:	0x0	IRQ from Input pins disabled
	0x1	IRQ from Input pins enabled

## INTSTAT2

---

Address: section "Device\_specific" base address + 0x0008

Description:

INTerrupt STATus

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

<b>IRQ_TIMER_END (7:0)</b>	
<i>RW2C</i>	This bit indicates that an interrupt has been detected on a Timer End event.
Value at Reset:	0x0

<b>IRQ_TIMER_START (7:0)</b>	
<i>RW2C</i>	This bit indicates that an interrupt has been detected on a Timer Start event.
Value at Reset:	0x0

**BUILDID**

Address: section "Device\_specific" base address + 0x001C

Description:  
This field represents a timestamp when the FPGA was synthesized and can be used to identify FPGA implementation uniquely.

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) RO	EPOCH date value	
Possible Values:	Any Value	



## FPGA\_ID

Address: section "Device\_specific" base address + 0x0020

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			PROFINET_LED	Reserved	PB_DEBUG_COM	Reserved	
7	6	5	4	3	2	1	0
Reserved			FPGA_ID(4:0)				

<b>PROFINET_LED</b>		
<i>RW</i>	This bit selects if the User Leds are controlled from the FPGA memory space, controlled by the host processor, or if the User Leds are controlled by the Microblaze running Profinet stack	
Value at Reset:	0x0	
Possible Values:	0x0	User Leds are under Host processor control
	0x1	User Leds are under Microblaze control

<b>PB_DEBUG_COM</b>		
<i>RW</i>	<p>These bits are used to redirect Profiblaaze UART output on internal COM port output. This should be used for debugging purposes only. The standard 0x3F8 COM1 on the SOC is connected to this physical line.</p> <p>Profiblaaze UART is hardcoded to 115200 bps, 8 data bit, no parity, 1 stop bit.</p> <p>This is effective on the Y7478-01 PCB only.</p>	
Value at Reset:	0x0	
Possible Values:	0x0	UART line between SOC and FPGA is tristated by the FPGA
	0x1	Profiblaaze output is seen on the internal com port

<b>FPGA_ID (4:0)</b>		
<i>RO</i>	This is the FPGA_ID.	
Possible Values:	0x1	Spartan6 LX9 fpga used on Y7449-00 (deprecated)
	0x2	Spartan6 LX16 fpga used on Y7449-01,02
	0x3	Artix7 A35T fpga used on Y7471-00 (deprecated)
	0x4	Artix7 A50T fpga used on Y7471-01
	0x5	Artix7 A50T fpga used on Y7471-02
	0x6	Artix7 A50T fpga used on Y7449-03
	0x7	Artix7 Spider PCIe on Advanced IO board
	0x8	Artix7 Ares PCIe (Iris3 Spider+Profiblaaze on Y7478-00)
	0x9	Artix7 Ares PCIe (Iris3 Spider+Profiblaaze on Y7478-01)
	0xA	Reserved for Artix7 Eris (LPC) on Y7478-01
	0x10	Iris GTX, Artix7 Ares PCIe, Artix7 A35T
	0x11	Iris GTX, Artix7 Ares PCIe, Artix7 A50T

## LED\_OVERRIDE

Address: section "Device\_specific" base address + 0x0024

Description:

This register is used to control LED override.

To signal a catastrophic condition, caused by thermal overrun, the BIOS uses this register to change the behavior of the USER LED. This register takes priority on every other register and setting controlling the LEDs

31	30	29	28	27	26	25	24
Reserved						RED_ORANG E_FLASH	ORANGE_OF F_FLASH
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							

<b>RED_ORANGE_FLASH</b>		
<i>RW</i>	When this bit is set, the LED will flash at approximately 1 Hz between the off state and the orange state	
Value at Reset:	0x0	
Possible Values:	0x0	Normal operation
	0x1	Flash override active

<b>ORANGE_OFF_FLASH</b>		
<i>RW</i>	When this bit is set, the LED will flash at approximately 1 Hz between the off state and the orange state	
Value at Reset:	0x0	
Possible Values:	0x0	Normal operation
	0x1	Flashing override active

## Section: INTERRUPT\_QUEUE

Address Range: [0x0040 - 0x0050]

Description:

This section controls the behavior of the interrupt queue

### CONTROL

Address: section "INTERRUPT\_QUEUE" base address + 0x0000

31	30	29	28	27	26	25	24
NB_DW(7:0)							
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							ENABLE

<b>NB_DW (7:0)</b> <i>STATIC</i>	
	This is the number of 32-bit DW used to represent all interrupt sources. It is used by the driver to know how to split the data of the interrupt queue in interrupt events.  This number should always be a power of 2 to simplify the hardware implementation and avoid having a single interrupt event split by the wrap-around boundary.
Value at Reset:	0x1

<b>ENABLE</b> <i>RW</i>	
	This bit is used to enable the interrupt queue. When disabled, the interrupt will behave in a legacy way where all interrupts are merged into interrupt status register and driver has to read the status register to know the interrupt sources.  To reset the interrupt queue, the driver should disable the queue and re-enable it. This will cause the producer index to be reset to 0 internally in the hardware. The driver should write the Producer index to 0 when the interrupt queue is disabled to prevent mis-interpretation of producer index when the queue is turned back to on.
Value at Reset:	0x0

## CONS\_IDX

Address: section "INTERRUPT\_QUEUE" base address + 0x0004

Description:

The consumer index indicates up to which element of interrupt queue array it can write. Element in the queue between CONS\_IDX (included) and PROD\_IDX (not included) belong to the driver and are not written by the hardware.

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

CONS_IDX (9:0) RW	
	When turning on the interrupt queue, the driver should first write this index to value 1023 (0X3FF) to indicate that the queue is empty.
Value at Reset:	0x0

## ADDR\_LOW

---

Address: section "INTERRUPT\_QUEUE" base address + 0x0008

Description:

This is the lower part of the address in host memory where the PCIe device writes the interrupt queue. It has to be aligned on 8K bytes boundary. The producer index is written at the end of the queue of 4K (ADDR\_LOW + 4KB).

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

<b>ADDR (31:0)</b> <i>RW (31:13)</i> <i>RO (12:0)</i>	
Value at Reset:	0x0

## ADDR\_HIGH

---

Address: section "INTERRUPT\_QUEUE" base address + 0x000C

Description:

This is the high part of the address in host memory where the PCIe device writes the interrupt queue. It must be written to 0 if the queue resides in the first 4 GB of memory.

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

<b>ADDR (31:0)</b> <i>RW</i>	
Value at Reset:	0x0

## MAPPING

**⚠ Note: Register readback is disabled.**

Address: section "INTERRUPT\_QUEUE" base address + 0x0010

Description:

This register is used to represent the mapping of the interrupt source in the event queue vector. All its bitfields have no effect when written and will read back to 0. Software should use the generated structure to identify the bit positions.

31	30	29	28	27	26	25	24
IRQ_TIMER_END(7:0)							
23	22	21	20	19	18	17	16
IRQ_TIMER_START(7:0)							
15	14	13	12	11	10	9	8
Reserved				IO_INTSTAT(3:0)			
7	6	5	4	3	2	1	0
Reserved		IRQ_TICK_LATCH	IRQ_MICROBLAZE	IRQ_TIMER	IRQ_TICK_WA	IRQ_TICK	IRQ_IO

<b>IRQ_TIMER_END (7:0)</b>	
<i>WO/AutoClr</i>	This bit indicates that an interrupt has been detected on a Timer End event.

<b>IRQ_TIMER_START (7:0)</b>	
<i>WO/AutoClr</i>	This bit indicates that an interrupt has been detected on a Timer Start event.

<b>IO_INTSTAT (3:0)</b>	
<i>WO/AutoClr</i>	This bit indicates that an interrupt has been detected on the corresponding I/O input.

<b>IRQ_TICK_LATCH</b>	
<i>WO/AutoClr</i>	This bit indicates that an interrupt has been detected on one of the latch associated with the matching tick table
Possible Values:	0x0 No interrupt detected
	0x1 Interrupt event occurred in the tick table

<b>IRQ_MICROBLAZE</b>	
<i>WO/AutoClr</i>	This bit indicates that the Microblaze subsystem has updated datapointer to exchange data with the host.

<b>IRQ_TIMER</b>	
<i>WO/AutoClr</i>	This bit indicates that an interrupt has been detected on one Timer, see INTSTAT2 register.
	This is a read-only register.

<b>IRQ_TICK_WA</b>	
<i>WO/AutoClr</i>	This bit indicates that a wrap around of the tick table X has happen.

<b>IRQ_TICK</b>	
<i>WO/AutoClr</i>	This bit indicates that an interrupt has been detected on the Tick Table X(half/full).

<b>IRQ_IO</b>	
<i>WO/AutoClr</i>	This bit indicates that an interrupt has been detected on the User Inputs.



## Section: SPI

Address Range: [0x00E0 - 0x00EC]

Description:

Cette section est laisse ici pour faire un fichier .H unifie mais les registre sous-jacent ne sont pas implante dans le FPGA.

### SPIREGIN

### SPI Register In

**⚠ Note: Register readback is disabled.**

Address: section "SPI" base address + 0x0000

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

<b>SPI_ENABLE</b> <i>WO</i>	<b>SPI ENABLE</b>	
	This bit enables the Output enable of the pin of the FPGA. This is needed to put the SPI interface in hi-Z when not using it.	
Value at Reset:	0x0	
Possible Values:	0x0	The SPI interface is disabled
	0x1	The SPI interface is enabled

<b>SPIRW</b> <i>WO</i>	<b>SPI Read Write</b>	
	Specify the SPI transfer type (read or write access).	
Value at Reset:	0x0	
Possible Values:	0x0	Write Access
	0x1	Read Access

<b>SPICMDDONE</b> <i>WO</i>	<b>SPI CoMmaD DONE</b>	
	Specify the last transaction for an SPI command sequence.	
Value at Reset:	0x0	

<b>SPISEL</b> <i>WO</i>	<b>SPI active channel SElection</b>	
	Selects the active SPI x channel. Current implementation uses a single channel to this field has a single bit.	
Value at Reset:	0x0	

<b>SPITXST</b>	SPI SPITXST Transfer STart
<i>WO/AutoClr</i>	Start an SPI transaction when 1 is written

  

<b>SPIDATAW (7:0)</b>	SPI Data byte to write
<i>WO</i>	This is the data byte to be written.
Value at Reset:	0x0

Address: section "SPI" base address + 0x0008

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

<b>SPI_WB_CAP</b>	SPI Write Burst CAPable	
<i>STATIC</i>	This register informs if the SPI core is able to write burst of 256 bytes to the SPI device (Write page), without requiring register polling between command, adress and data bytes in the write page command.	
Value at Reset:	0x0	
Possible Values:	0x0	This fpga can't do write burst
	0x1	This fpga is capable of doing write burst

<b>SPIWRTD</b>	SPI Write or Read Transfer Done	
<i>STATIC</i>	Specify if there is a transfer in progress.	
Value at Reset:	0x0	
Possible Values:	0x0	Transfer in progress
	0x1	No transfer in progress

<b>SPIDATARD (7:0)</b>	SPI DATA Read byte OUTput	
<i>STATIC</i>	This is the data read byte from the SPI	
Value at Reset:	0x0	

## Section: arbiter

Address Range: [0x00F0 - 0x00F8]

### ARBITER CAPABILITIES

Address: section "arbiter" base address + 0x0000

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

AGENT_NB (1:0)	
STATIC	Number of agents
Value at Reset:	0x2

TAG (11:0)	
STATIC	Arbiter TAG : AABiter
Value at Reset:	0xAAB

## AGENT (1:0)

Address: section "arbiter" base address + 0x0004 + (index \* 0x4)

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						ACK	REC
7	6	5	4	3	2	1	0
Reserved			DONE	Reserved			REQ

<b>ACK</b> <i>RO</i>	master request ACKnowledge	
	This field indicates that a master request has been received, and the resource is ready to be used.	
Possible Values:	0x0	The resource is NOT ready to be used.
	0x1	The resource is ready to be used.

<b>REC</b> <i>RO</i>	master request RECeived	
	This field indicates that a master request has been received.	
Possible Values:	0x0	Master request not received
	0x1	Master request has been received

<b>DONE</b> <i>WO/AutoClr</i>	transaction DONE	
	This field from master requester informs the arbiter that it has finish with the device resource. The arbiter can give the resource to another master requester.	
Possible Values:	0x0	Nothing
	0x1	Master requester transaction done

<b>REQ</b> <i>WO/AutoClr</i>	REQuest resource	
	This field from master requester ask the arbiter for a device resource.	
Possible Values:	0x0	Nothing
	0x1	Ask for a device resource

## Section: IO (1 : 0)

Address Range: [0x0200 - 0x027C]

Section repeated 2 times. IO(i) base address located @ 0x0200 + (i \* 0x80)

Description:

This sections are for IO banks used to interface to external world.

Bank0 is UserInput

Bank1 is UserOutput

## CAPABILITIES\_IO

Address: section "IO" base address + 0x0000

Description:

This register identifies this block as a IO block for software automatic feature detection mechanism

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

<b>IO_ID (7:0)</b> <i>STATIC</i>							
	This identifies an input and/or output bank.  This is a Legacy module. The FREV field has been removed to made the ID larger (8 bits). If for an unknown reason this module have to be modified, change the ID to a new one, and add new fonctionnality.						
Value at Reset:	0x10						

<b>N_port (4:0)</b> <i>RO</i>							
	This is the number of bits in the 'bank'. This counter is 0 based. A value of 0 indicates 1 bit and the value of 1fh indicates 32 bits.						

<b>Input</b> <i>RO</i>							
	Indicates if input is available on this feature port.						
Possible Values:	0x0	No input capabilities					
	0x1	Input capabilities present					

<b>Output</b> <i>RO</i>		
	Indicates if output is available on this feature port.	
Possible Values:	0x0	No output capabilities
	0x1	Output capabilities present

<b>Intnum (4:0)</b> <i>RO</i>		
	This is the bit number in the interrupt field when the interrupts from this I/O banks are forwarded. When an I/O generates an interrupt, the corresponding bit will be set in the IO_INTSTAT register in the I/O bank. Also, an interrupt will be forwarded to the global interrupt register of the FPGA.	

## IO PIN

---

Address: section "IO" base address + 0x0004

Description:

This register is present only if the I/O bank is input capable.

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				Pin_value(3:0)			

<b>Pin_value (3:0)</b>		
<i>RO</i>	This bit is the status of the pin. The value read back in is not influenced by the io_pol bit.	
Possible Values:	0x0	Input is de-asserted
	0x1	Input is asserted



## IO\_OUT

---

Address: section "IO" base address + 0x0008

Description:

This register is present only if the I/O bank is output capable.

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				Out_value(3:0)			

<b>Out_value (3:0)</b>		
<i>RW</i>	This bit controls the output bit, when out_sel is set to 0.	
Value at Reset:	0x0	
Possible Values:	0x0	Output will be low
	0x1	Output will be high

## IO DIR

Address: section "IO" base address + 0x000C

Description:

This register is present only if the I/O bank is output capable.

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				Dir(3:0)			

<b>Dir (3:0)</b>		
<i>RW</i>	This bit controls if the user bit is an output or an input.	
Value at Reset:	0x0	
Possible Values:	0x0	User pin is an input
	0x1	User pin is an output. Input functions still work (interrupt, readback) but the input value will be the driven value.

## IO\_POL

Address: section "IO" base address + 0x0010

Description:

User bit polarity

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				In_pol(3:0)			

<b>In_pol (3:0)</b>		
<i>RW</i>	This controls the active state on the input path from the pin to the other component of the user I/O (interrupt)	
Value at Reset:	0x0	
Possible Values:	0x0	User I/O input generates an interrupt on rising edge.
	0x1	User I/O input generates an interrupt on falling edge.

## IO\_INTSTAT

Address: section "IO" base address + 0x0014

### Description:

Every bit in this register represents the status of an interrupt. When an event occurs, the corresponding bit is set in this register. If the output of the interrupt is enabled, the PCI interrupt pin will be asserted. The software should read this register when there is an interrupt. When a '1' is written in a bit, that event is acknowledged and the bit is returned to '0', unless the event is still occurring. When a '0' is written, nothing happens.

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				Intstat(3:0)			

<b>Intstat (3:0)</b>		
<i>RW2C</i>	This bit indicates that an interrupt has been detected on the corresponding I/O input.	
Value at Reset:	0x0	
Possible Values:	0x0	No interrupt detected
	0x1	Interrupt event occurred

## IO\_INTMASKn

Address: section "IO" base address + 0x0018

Description:

Every bit in this register is used to mask some event. While a bit in this register is set to 0, the corresponding bit in the INTSTAT register cannot SWITCH to 1. When a bit in the INTMASK register is set to '1', the corresponding bit in INTSTAT function normally

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				Intmaskn(3:0)			

<b>Intmaskn (3:0)</b>		
<i>RW</i>	Interrupt IRQ MASK not. When set to 0, the associated IRQ will be not generated.	
Value at Reset:	0x0	
Possible Values:	0x0	No interrupt will be generated for the corresponding user I/O. (Note that if the corresponding bit is already asserted in the IO_INTSTAT register, the interrupt will still be generated)
	0x1	Interrupt generated when the corresponding I/O toggles with the polarity defined in IO_POL register.

## IO\_ANYEDGE

---

Address: section "IO" base address + 0x001C

Description:

Generate an interrupt on any edge on the input signal

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				In_AnyEdge(3:0)			

<b>In_AnyEdge (3:0)</b> <i>RW</i>	
	This bit is used to override the In_pol setting bit of the corresponding register. When this bit is set, an interrupt is generated on rising and falling edge of the input signal. When this bit is 0, the code work in legacy mode and the in_pol is used to select which edge generate an interrupt.  There is one bit per input signal.
Value at Reset:	0x0

## Section: Quadrature (0 : 0)

---

Address Range: [0x0300 - 0x037C]

Section repeated 1 times. Quadrature(i) base address located @ 0x0300 + (i \* 0x80)

Description:

This section controls a single quadrature decoder.

### CAPABILITIES\_QUAD

---

Address: section "Quadrature" base address + 0x0000

Description:

This register identifies the capabilities of the Quadrature of this feature section.

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

<b>QUADRATURE_ID (7:0)</b>	
<i>STATIC</i>	Any feature with 8 MSB set to 0x64 is a ticktable as defined in the following registers.
Value at Reset:	0x64

<b>FEATURE_REV (3:0)</b>	
<i>STATIC</i>	Revision of the feature. This field must be used by software to detect if the current software support the register definition of this feature.
Value at Reset:	0x0

## PositionReset

Address: section "Quadrature" base address + 0x0004

Description:

Describes the event that can reset the position of the quadrature decoder

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		PositionResetSource(3:0)				PositionReset Activation	soft_PositionR eset

<b>PositionResetSource (3:0)</b> <i>RW</i>		
	Quadrature Position counter Reset Source.	
	The Reset Source set to Disable only affects the reset from the Input Line and the reset at trigger position.	
	The soft_PositionReset may be executed even if PositionResetSource is disable.	
Value at Reset:	0x0	
Possible Values:	0x0	Disable Reset source
	0x1 - 0x4	Input Line
	0x5	Counter reaches position trigger

<b>PositionResetActivation</b> <i>RW</i>		
	Quadrature Position counter Reset activation edge for line inputs.	
Value at Reset:	0x0	
Possible Values:	0x0	Rising edge
	0x1	Falling edge

<b>soft_PositionReset</b> <i>WO/AutoClr</i>		
	This is the quadrature counter software reset. Set this field reset the counters in the quadrature logic to 0.	
Possible Values:	0x0	Nothing
	0x1	Software position reset



## DecoderInput

Address: section "Quadrature" base address + 0x0008

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

<b>BSelector (2:0)</b>		
<i>RW</i>	This is the phase B line. By definition, this line transitions BEFORE phase A line when the encoder is rotating in the clockwise direction.	
Value at Reset:	0x2	
Possible Values:	0x0 - 0x3	Input Line

<b>ASelector (2:0)</b>		
<i>RW</i>	This is the phase A line. By definition, this line transitions AFTER phase B line when the encoder is rotating in the clockwise direction.	
Value at Reset:	0x1	
Possible Values:	0x0 - 0x3	Input Line

## DecoderCfg

Address: section "Quadrature" base address + 0x000C

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			DecOutSource0(2:0)			Reserved	QuadEnable

<b>DecOutSource0 (2:0)</b> <i>RW</i>		
	Quadrature Output Source for output 0 of the decoder. Selects wich signal of the rotary decoder is sent to the output 0 of the core.  There is only one output per decoder at this moment. This may change on customer needs.	
Value at Reset:	0x0	
Possible Values:	0x0	New Tick
	0x1	Clock Wise Tick
	0x2	Counter Clock Wise Tick
	0x3	Any Tick (Clock or counter clock wise ticks)
	0x4	Counter reaches Position Trigger register(Regenerated Tick)

<b>QuadEnable</b> <i>RW</i>		
	This register enables the Quadrature decoder.  When set to disable, the position/maximum position/direction is also reseted.	
Value at Reset:	0x0	
Possible Values:	0x0	Quadrature decoder disable
	0x1	Quadratue decoder enable

## DecoderPosTrigger

---

Address: section "Quadrature" base address + 0x0010

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

<b>PositionTrigger (31:0)</b>		
<i>RW</i>	Position counter where the decoder will generate an event(or a regeretated Tick). When routing this event to the RstSource, the logic can divide the rotary ticks by a factor.	
Value at Reset:	0x1	
Possible Values:	0x1 - 0xFFFFFFFF	Any 32 bits value in this range

## DecoderCntrLatch\_Cfg

Address: section "Quadrature" base address + 0x0014

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

<b>DecoderCntrLatch_SW</b> <i>WO/AutoClr</i>							
	This field register is used to enable the copy the actual quad decoder counter into the DecoderCntrLatched_SW register.  This is used to compensate for the size of the actual decoder counter which is larger than an atomic read.						
Possible Values:	0x0	Nothing					
	0x1	Latch the quad decoder counter into DecoderCntrLatched_SW register.					

<b>DecoderCntrLatch_Src (4:0)</b> <i>RW</i>							
	This field is used to select which of the input signal is used to latch the current quad decoder counter into register DecoderCntrLatched.						
Value at Reset:	0x0						
Possible Values:	0x0 - 0x3	Input Line					
	0x4 - 0x6	Internal input line					
	0x7 - 0xE	Timer Output					

<b>DecoderCntrLatch_En</b> <i>RW</i>							
	This field register is used to enable the copy the actual quad decoder counter into the DecoderCntrLatched register when an input transitioned.  This is used to compensate for the size of the actual decoder counter which is larger than an atomic read.						
Value at Reset:	0x0						
Possible Values:	0x0	Nothing					
	0x1	Enable the Quad decoder counter latch from Inputs and Timers					

<b>DecoderCntrLatch_Act (1:0)</b> <i>RW</i>							
	Specify which edge of the input signal is used to copy the current quad decoder counter into a register.						
Value at Reset:	0x0						
Possible Values:	0x0	RisingEdge					
	0x1	FallingEdge					
	0x2	AnyEdge					
	0x3	None (edge detection disable)					



**DecoderCntrLatched\_SW**

Address: section "Quadrature" base address + 0x0034

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

<b>DecoderCntr (31:0)</b> <i>RO</i>		
	This is the Quad decoder counter latched with the SW snapshot.	
Possible Values:	Any Value	

# DecoderCntrLatched

Address: section "Quadrature" base address + 0x0038

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

DecoderCntr (31:0)		
RO	This is the Quad decoder counter latched with the Input or Timer source.	
Possible Values:	Any Value	

## Section: TickTable (0 : 0)

Address Range: [0x0380 - 0x03FC]

Section repeated 1 times. TickTable(i) base address located @ 0x0380 + (i \* 0x80)

### CAPABILITIES\_TICKTBL

Address: section "TickTable" base address + 0x0000

Description:

This register identifies the capabilities of the Tick table interface of this feature section.

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

<b>TICKTABLE_ID (7:0)</b>	
<i>STATIC</i>	Any feature with 8 MSB set to 0x61 is a ticktable as defined in the following registers.
Value at Reset:	0x61

<b>FEATURE_REV (3:0)</b>	
<i>STATIC</i>	Revision of the feature. This field must be used by software to detect if the current software support the register definition of this feature.  Revision 1 is used to notify the software that the structure of the following register has changed
Value at Reset:	0x1

<b>NB_ELEMENTS (4:0)</b>	
<i>STATIC</i>	This field is used to present the number of elements in the tick table, represented in power of 2. For example, a value of 8 is for 256 element in the table, a value of 13 is for 8192 elements in the table,
Value at Reset:	0xd
Possible Values:	0x4 - 0x1F

<b>INTNUM (4:0)</b>	
<i>RO</i>	This is the bit number in the global interrupt status where interrupt from this feature is forwarded. Periodic interrupt is automatically generated at every half table crossing whenever the EnableHalftableInt is turned on.



## CAPABILITIES\_EXT1

Address: section "TickTable" base address + 0x0004

Description:

This is the extension of the capabilities register. Its bit mapping content is Feature\_rev specific.

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				NB_LATCH(3:0)			
7	6	5	4	3	2	1	0
TABLE_WIDTH(7:0)							

<b>NB_LATCH (3:0)</b>	
<i>STATIC</i>	This is the number of latch. This counter is 1 based. A value of 1 indicates 1 latch and the value of 0x0Ch indicates 12 latches
Value at Reset:	0x2

<b>TABLE_WIDTH (7:0)</b>	
<i>STATIC</i>	This is the width of the tick table, as seen by software. This counter is 1 based.
Value at Reset:	0x4

TickTableClockPeriod

Address: section "TickTable" base address + 0x0008

Description:

This is the period of the internal clock used by the timer.

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
Period_ns(7:0)							

Period_ns (7:0)	
RO	Period in ns of the system reference clock for generate the clock used by the Ticktable (see register IntClock_sel). In Gpm Ref, clock is LPC clock at T=30ns (33.333Mhz). In Gpm-Atom, Ref clock is LPC clock at T=40ns (25.000Mhz). In Ares and Spider PCIe, Ref clock is PCIe clock at T=16ns (62.500Mhz).

## TickConfig

Address: section "TickTable" base address + 0x000C

31	30	29	28	27	26	25	24
Reserved			ClearTickTable	Reserved			
23	22	21	20	19	18	17	16
ClearMask(7:0)							
15	14	13	12	11	10	9	8
Reserved				TickClock(3:0)			
7	6	5	4	3	2	1	0
IntClock_sel(1:0)		TickClockActivation(1:0)		EnableHalfTableInt	IntClock_en	LatchCurrentStamp	ResetTimestamp

<b>ClearTickTable</b> <i>WO/AutoClr</i>	Clear command in Tick Table
	This field is used to clear the command contents of the ticktable without reset all the core. The commands bits in the ticktable to clear are configured in field ClearMask. The operation will take around XXXX us.

<b>ClearMask (7:0)</b> <i>RW</i>	Clear command Mask
	This field is used to mask bits in the ticktable when a ClearTickTable command is sent. When the associated mask bit is set to '1' the bit command will not be clear to 0 (Do not change output)
Value at Reset:	0x0

<b>TickClock (3:0)</b> <i>RW</i>		
	This field is used to select the clock source.  When select Internal clock as source, select corresponding activation register to rising edge (0x0).	
Value at Reset:	0x0	
Possible Values:	0x0	Internal clock source
	0x1 - 0x4	Input Line
	0x5	QuadratureDecoder Output

<b>IntClock_sel (1:0)</b> <i>RW</i>		
	This register selects the Clock Int frequency used in the logic.  IntClock_sel = 0x0 : T=Period_ns * 8192 IntClock_sel = 0x1 : T=Period_ns * 2048 IntClock_sel = 0x2 : T=Period_ns * 1024 IntClock_sel = 0x3 : T=Period_ns * 256	
Value at Reset:	0x1	
Possible Values:	0x0	T=Period_ns * 8192 : Clock Int is 4.069 Khz(GPm), 3.05175 Khz(GPm-Atom), 7.629 Khz(Spider_PCIE)
	0x1	T=Period_ns * 2048 : Clock Int is 16.276 Khz(GPm), 12.207 Khz(GPm-Atom), 30.518Khz(Spider_PCIE) (Default)
	0x2	T=Period_ns * 1024 : Clock Int is 32.552 Khz(GPm), 24.414 Khz(GPm-Atom), 61.035Khz(Spider_PCIE)
	0x3	T=Period_ns * 256 : Clock Int is 130.208 Khz(GPm), 97.656 Khz(GPm-Atom), 244.141Khz(Spider_PCIE)

<b>TickClockActivation (1:0)</b>		
<i>RW</i>	Specify which edge of the input signal is used to clock the current timestamp.	
Value at Reset:	0x0	
Possible Values:	0x0	RisingEdge
	0x1	FallingEdge
	0x2	AnyEdge
	0x3	None (edge detection disable)

<b>EnableHalftableInt</b>		
<i>RW</i>	This bit turns on the interrupt mechanism to trigger the refill of the tick table from the software backup list. An interrupt is generated when the first half or the second half of the tick table just completed execution. Upon receipt of this interrupt, the software can latch and read the current timestamp to determine if it can rewrite the first or second half of the ticktable.	
Value at Reset:	0x0	
Possible Values:	0x0	No interrupt are generated
	0x1	An interrupt is generated whenever an half of the table has been executed(first half and second half).

<b>IntClock_en</b>		
<i>RW</i>	Internal Clock enable	
Value at Reset:	0x0	
Possible Values:	0x0	Internal clock disabled
	0x1	Internal clock enabled

<b>LatchCurrentStamp</b>		
<i>WO/AutoClr</i>	This bit is used to copy the actual timestamp into the CurrentStampLatched register. This is used to compensate for the size of the actual timestamp which is larger than an atomic read.	
Possible Values:	0x0	Nothing
	0x1	Latch current stamp to register

<b>ResetTimestamp</b>		
<i>WO/AutoClr</i>	This field is used to reset the CurrentStamp register. It will also erase the full content of the associated ticktable. The operation will take around 250us.	
Possible Values:	0x0	Nothing
	0x1	Reset ticktable and counters

# CurrentStampLatched

Address: section "TickTable" base address + 0x0010

Description:

This is the value of the current timestamp. Software must write '1' in the LatchCurrentStamp field in the control register to copy the actual timestamp into this register because the actual timestamp is too large to be read in an atomic access.

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

<b>CurrentStamp (31:0)</b> <i>RO</i>		
	This number is the current "time" stamp for the table when LatchCurrentStamp register was set. When the table is clocked from external source, the time is measured in number of external ticks.	
	The LSB of this timestamp is the index used in the table. The NB_ELEMENTS field of the capabilities register can be used to determine the number of LSB used as an index in the table. The other MSB can be used as a lap counter.	
	This value can be reset through register ResetTimestamp.	
Possible Values:	0x0 - 0xFFFFFFFF	Any 32 bits value

# WriteTime

Address: section "TickTable" base address + 0x0014

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

<b>WriteTime (31:0)</b> <i>RW</i>		
	This is the time where the next command will be written. The LSBs are used to select where in the tick table the command is written.	
	The whole write time is also compare with the actual current timestamp to determine if the requested write time is in the future or in the past. If the requested command time is in the past, the command write is discarded. The hardware automatically handles the wraparound by substracting the current time from the write time; if the result is positive, the command is in the future.	
	Note that if a command is written where the WriteTime is exactly equal to the current timestamp, it is considered to be in the past.	
Value at Reset:	0x0	
Possible Values:	0x0 - 0xFFFFFFFF	Any 32 bits value

## WriteCommand

Address: section "TickTable" base address + 0x0018

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		WriteDone	WriteStatus	Reserved		ExecuteFutureWrite	ExecuteImmWrite
7	6	5	4	3	2	1	0
Reserved	BitCmd(1:0)		Reserved			BitNum(1:0)	

<b>WriteDone</b> <i>RO</i>							
	This bit field informs the driver that the previous cmd has been executed and the WriteStatus field is updated. This bit is reset when the command is received.						
Possible Values:	0x0	Last Cmd running					
	0x1	Last Cmd executed					

<b>WriteStatus</b> <i>RO</i>							
	This updated upon a write to the ExecuteFutureWrite bit. It is undefined before that.						
Possible Values:	0x0	Last ExecuteFutureWrite resulted in failure					
	0x1	Last ExecuteFutureWrite resulted in success					

<b>ExecuteFutureWrite</b> <i>WO/AutoClr</i>							
	This triggers the write in the table. This will store the BitCmd in the associate BitNum if the WriteTime is in the future. Write status is reflected in the WriteStatus field.						
Possible Values:	0x0	Nothing					
	0x1	Future write snapshot					

<b>ExecuteImmWrite</b> <i>WO/AutoClr</i>							
	This triggers the write in the table. This will store the BitCmd in the associate BitNum in the next entry in the tick table to be executed, thereby ignoring the WriteTime field. This always succeeds						
Possible Values:	0x0	Nothing					
	0x1	Imminent write snapshot					

<b>BitCmd (1:0)</b> <i>RW</i>							
	This is the bit command to insert in the table to affect the virtual output of the tick table. The Rise-then-fall will generate a very short pulse and is intended to trigger another submodule internal to the Spider.						
Value at Reset:	0x0						
Possible Values:	0x0	Do not change output					
	0x1	Rise output					
	0x2	Fall output					
	0x3	Rise-then-fall output.					

<b>BitNum (1:0)</b>		
<i>RW</i>	This is the bit number affected by this write command.	
Value at Reset:	0x0	
Possible Values:	0x0 - 0x3	Bit number affected by the command



## LatchIntStat

Address: section "TickTable" base address + 0x001C

Description:

This register holds the status of the interrupts of the latches. There are as many bits as there are latches.

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						LatchIntStat(1:0)	

<b>LatchIntStat (1:0)</b> <i>RW2C</i>		
	This bit indicates that the InputStamp condition has occurred and an updated value (which could be the same as the previous one) is available in corresponding InputStampLatched register.	
Value at Reset:	0x0	
Possible Values:	0x0	No interrupt condition
	0x1	Interrupt condition occurred

## InputStamp (1:0)

Address: section "TickTable" base address + 0x0020 + (index \* 0x4)

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

<b>InputStampSource (3:0)</b>		
<i>RW</i>		This field is used to select which of the input signal is used to latch the current ticktable timestamp into a register. Note that software must write LatchInputStamp field to make this register indirectly readable through InputStampLatched register.
Value at Reset:		0x0
Possible Values:	0x0 - 0x3	Input Line
	0x4 - 0x6	Internal input line
	0x7 - 0xE	Timer Output

<b>LatchInputIntEnable</b>		
<i>RW</i>		This field register is used to enable the interrupt generation when a latch event occurs. When the latch condition occurs, as determined by the InputStampSource, InputStampActivation and LatchInputStamp_en, an interrupt will be generated if it is enabled by this register.
Value at Reset:		0x0
Possible Values:	0x0	Interrupt generation disabled
	0x1	Interrupt generation enabled

<b>LatchInputStamp_En</b>		
<i>RW</i>		This field register is used to enable the copy the actual timestamp into the InputStampLatched register when an input transitioned.  This is also used to compensate for the size of the actual timestamp which is larger than an atomic read.
Value at Reset:		0x0
Possible Values:	0x0	Nothing
	0x1	Enable the Input stamp arm logic

<b>InputStampActivation (1:0)</b>		
<i>RW</i>		Specify which edge of the input signal is used to copy the current timestamp into a register.
Value at Reset:		0x0
Possible Values:	0x0	RisingEdge
	0x1	FallingEdge
	0x2	AnyEdge
	0x3	None (edge detection disable)

**reserved\_for\_extra\_latch (9:0)**

Address: section "TickTable" base address + 0x0028 + (index \* 0x4)

Description:

This register is used to take space for extra latches. This pushes the InputStampLatched register array at offset +0x50 from the feature ID.

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							reserved_for_e xtra_latch

reserved_for_extra_latch	
STATIC	
Value at Reset:	0x0

## InputStampLatched (1:0)

Address: section "TickTable" base address + 0x0050 + (index \* 0x4)

Description:

This is the value of the timestamp saved when an input transitioned. Software must write '1' in the LatchInputStamp field in the control register to copy the actual timestamp into this register because the actual timestamp is too large to be read in an atomic access.

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

<b>InputStamp (31:0)</b> <i>RO</i>		
	This number is the "time" stamp saved when an input transitioned(selected by register InputStampSource). When the table is clocked from external source, the time is measured in number of external ticks.	
	This value can be reset through register ResetTimestamp.	
Possible Values:	0x0 - 0xFFFFFFFF	Any 32 bits value

## Section: InputConditioning

Address Range: [0x0400 - 0x047C]

Description:

This section refers to every physical input line

### CAPABILITIES\_INCOND

Address: section "InputConditioning" base address + 0x0000

Description:

This register identifies the capabilities of the input conditioning of this feature.

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

<b>INPUTCOND_ID (7:0)</b>	
<i>STATIC</i>	Any feature with 8 MSB set to 0x62 is a the line input conditioning configuration as defined in the following registers.
Value at Reset:	0x62

<b>FEATURE_REV (3:0)</b>	
<i>STATIC</i>	Revision of the feature. This field must be used by software to detect if the current software support the register definition of this feature.
Value at Reset:	0x0

<b>NB_INPUTS (4:0)</b>	
<i>STATIC</i>	This is the number of inputs controlled by this feature, which is also the number of register following this capabilities register.
Value at Reset:	0x4

<b>Period_ns (7:0)</b>	
<i>RO</i>	Period in ns of the internal clock reference by default. In the GPM, Ref clock is LPC clock at T=30ns (0x1e ns) In the GPM-Atom, Ref clock is LPC clock at T=40ns (0x28 ns) In Spider PCIe and Ares, Ref clock is PCIe clock at T=16ns (62.500Mhz).
Possible Values:	Any Value      Any 8 bits value



## InputConditioning (3:0)

Address: section "InputConditioning" base address + 0x0004 + (index \* 0x4)

Description:

Every bit in this register is used to invert signal polarity.

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

<b>DebounceHoldOff (23:0)</b>			
<i>RW</i>	<p>This is the minimal time from a valid input signal edge to the next edge. After a valid edge is detected, any other edge is considered noise and is suppressed until the time period defined by this field is elapsed.</p> <p>The reference clock for this feature is the system clock.</p> <p>For GPM IvB: the clock used is T=30ns, f=33.3333Mhz, it gives 0.5ms of maximal debounce hold off.</p> <p>For GPM ByT: the clock used is T=40ns, f=25.0000Mhz, it gives 0.6ms of maximal debounce hold off.</p> <p>For Spider PCIe (Indio): the clock used is T=16ns, f=62.500Mhz, it gives 0.26ms of maximal debounce hold off.</p> <p>For GTR PCB rev 00: the clock used is T= 16ns.</p> <p>For GTR PCB rev 01: the clock used is T= 40 ns.</p>		
Value at Reset:	0x0		
Possible Values:	0x0 - 0xFFFFFFFF	Any 24 bits value	

<b>InputFiltering</b>			
<i>RW</i>	Input pulse shorter		
Value at Reset:	0x0		
Possible Values:	0x0	Filtering OFF	
	0x1	500 ns +/- 10% filtering enabled	

<b>InputPol</b>			
<i>RW</i>	<p>This register set the input polarity.</p> <p>When set to '0' the polarity on the pin of the fpga is not inverted to the core.</p>		
Value at Reset:	0x0		
Possible Values:	0x0	Not invert polarity	
	0x1	Invert polarity	

## Section: OutputConditioning

Address Range: [0x0480 - 0x04FC]

Description:

This section is associated with every output

### CAPABILITIES\_OUTCOND

Address: section "OutputConditioning" base address + 0x0000

Description:

This register identifies the capabilities of the output conditionning module of this feature.

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

<b>OUTPUTCOND_ID (7:0)</b>	
<i>STATIC</i>	Any feature with 8 MSB set to 0x63 is a line input conditionning configuration as defined in the following registers.
Value at Reset:	0x63

<b>FEATURE_REV (3:0)</b>	
<i>STATIC</i>	Revision of the feature. This field must be used by software to detect if the current software support the register definition of this feature.
Value at Reset:	0x0

<b>NB_OUTPUTS (4:0)</b>	
<i>STATIC</i>	This is the number of outputs controlled by this feature, which is also the number of register following this capabilities register.
Value at Reset:	0x4



## OutputCond (3:0)

Address: section "OutputConditioning" base address + 0x0004 + (index \* 0x4)

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

<b>OutputVal</b>	Output Value
<i>RO</i>	This is logical value on the FPGA USEROUT pin, after the output filter

<b>OutputPol</b>		
<i>RW</i>	This register set the output polarity. When set to '0' the polarity on the output pin of the fpga is the same as in the core.	
Value at Reset:	0x0	
Possible Values:	0x0	Do not change polarity
	0x1	Polarity inverted

<b>Outsel (5:0)</b>		
<i>RW</i>	Selects what is output	
Value at Reset:	0x0	
Possible Values:	0x0	Static output programmed in IO module
	0x1 - 0x4	Tick Table
	0x5	QuadratureDecoder
	0x6 - 0xD	Timer Output
	0xE - 0x10	Internal inputs

**Reserved**

Address: section "OutputConditioning" base address + 0x0014

Description:  
Reserved space for future use

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(7:0)							

Reserved (7:0) <i>STATIC</i>		
	Reserved for future use	
Value at Reset:	0x0	
Possible Values:	0x0	No interrupt detected
	0x1	Interrupt event occurred

## Output Debounce

Address: section "OutputConditioning" base address + 0x002C

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

<b>Output_HoldOFF_reg_EN</b>	
<i>RW</i>	To limit the output rate of the UserOutputs, the Output Debounce logic is always enable and limited by HW fpga to 61Khz. In case we want to modify this output toggle rate in the near future, to increase the toggle rate of the outputs, the driver can set this register (Output_HoldOFF_reg_EN) to '1' and program the register Output_HoldOFF_reg_CNTR to the desired value.
Value at Reset:	0x0

<b>Output_HoldOFF_reg_CNTR (9:0)</b>	
<i>RW</i>	<p>This is the minimal time from a valid output signal edge to the next edge. After a valid edge is detected, any other edge is suppressed until the time period defined by this field is elapsed. At the end of this time, the level of the output is compared with the internal signal, and if it's differs, the output is updated with the new value and the timer is restarted. If they are equal, the logic will wait for an edge of the internal signal.</p> <p>For limit the output rate follow this simple formula :</p> $\text{Output\_HoldOFF\_reg\_CNTR} = 1 / (2 * F\_lim * T)$ <p>For GPM: The clock used is <math>f=33.3333\text{Mhz}</math>, <math>T=30\text{ns}</math>.  For GPM Atom: The clock used is <math>f=25.0000\text{Mhz}</math>, <math>T=40\text{ns}</math>.</p> <p>For a limit of <math>F\_lim=61\text{Khz}</math> on the outputs :</p> <p>For the GPM the value of the register is : 273 dec  For the GPM Atom the value of the register is : 205 dec</p>
Value at Reset:	0x1ff

## Section: InternalInput

Address Range: [0x0500 - 0x057C]

Description:

This section is to configure some of the inputs of the FPGA that are kept internal in the product but external to the FPGA because of partitioning of the logic into many FPGA for hardware design. The first use of this feature is to define the following internal input from the Athena grab FPGA:

acq\_exposure        connecte sur internal\_input(0)  
acq\_strobe         connecte sur internal\_input(1)  
acq\_trigger\_ready connecte sur internal\_input(2)

### CAPABILITIES\_INT\_INP

Address: section "InternalInput" base address + 0x0000

Description:

This register identifies the capabilities of the internal output module of this feature.

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

<b>INT_INPUT_ID (7:0)</b>	
<i>STATIC</i>	Any feature with 8 MSB set to 0x66 is an internal input as defined in the following registers.
Value at Reset:	0x66

<b>FEATURE_REV (3:0)</b>	
<i>STATIC</i>	Revision of the feature. This field must be used by software to detect if the current software support the register definition of this feature.
Value at Reset:	0x0

<b>NB_INPUTS (4:0)</b>	
<i>STATIC</i>	This is the number of internal inputs. This number is solely here to allow software to generate the enumeration values of the various MUX used in the product.
Value at Reset:	0x3

## Section: InternalOutput

Address Range: [0x0580 - 0x05FC]

Description:

This section is to configure some of the outputs of the FPGA that are kept internal in the product but external to the FPGA because of partitioning of the logic into many FPGA for hardware design. The first use of this feature is to connect the trigger source to the trigger signal sent to the acquisition FPGA inside the Iris GTR.

### CAPABILITIES\_INTOUT

Address: section "InternalOutput" base address + 0x0000

Description:

This register identifies the capabilities of the internal output module of this feature.

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

<b>INT_OUTPUT_ID (7:0)</b>	
<i>STATIC</i>	Any feature with 8 MSB set to 0x65 is the line input conditioning configuration as defined in the following registers.
Value at Reset:	0x65

<b>FEATURE_REV (3:0)</b>	
<i>STATIC</i>	Revision of the feature. This field must be used by software to detect if the current software support the register definition of this feature.
Value at Reset:	0x0

<b>NB_OUTPUTS (4:0)</b>	
<i>STATIC</i>	This is the number of outputs controlled by this feature, which is also the number of register following this capabilities register.
Value at Reset:	0x1

## OutputCond (0:0)

Address: section "InternalOutput" base address + 0x0004

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

<b>OutputVal</b> <i>RO</i>	Output Value
	This is logical value on the FPGA USEROUT pin, after the output filter
	This register is for test purposes only and should not be used by the driver.

<b>Outsel (5:0)</b> <i>RW</i>		
	Selects what is output	
Value at Reset:	0x0	
Possible Values:	0x0 - 0x3	Tick Table
	0x4	QuadratureDecoder
	0x5 - 0xC	Timer Output
	0xD - 0x10	Line Input
	0x11	Microblaze internal output

## Section: Timer (7 : 0)

Address Range: [0x0600 - 0x067C]

Section repeated 8 times. Timer(i) base address located @ 0x0600 + (i \* 0x80)

### CAPABILITIES\_TIMER

Address: section "Timer" base address + 0x0000

Description:

This register identifies the capabilities of the timer interfaces to this feature section

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

<b>TIMER_ID (7:0)</b>	
<i>STATIC</i>	Any feature with 8 MSB set to 0x60 is a timer as defined in the following registers.
Value at Reset:	0x60

<b>FEATURE_REV (3:0)</b>	
<i>STATIC</i>	Revision of the feature. This field must be used by software to detect if the current software support the register definition of this feature.
Value at Reset:	0x0

<b>INTNUM (4:0)</b>	
<i>RO</i>	This is the bit number in the global interrupt status where interrupt from this feature is forwarded. Periodic interrupt is automatically generated at every half table crossing whenever the EnableHalfTableInt is turned on.

# TimerClockPeriod

Address: section "Timer" base address + 0x0004

Description:

This is the period of the internal clock used by the timer.

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
Period_ns(15:8)							
7	6	5	4	3	2	1	0
Period_ns(7:0)							

<b>Period_ns (15:0)</b> <i>RO</i>	Period in ns of the internal clock reference by default. In the GPM, Ref clock is LPC clock at T=30ns. Timer clock is 30x8=240ns (0xf0 ns). In the GPM-Atom, Ref clock is LPC clock at T=40ns. Timer clock is 40x8=320ns (0x140 ns). In Spider PCIe and Ares, Ref clock is PCIe clock at T=16ns. Timer clock is 16x8=128ns (0x80 ns).	
Possible Values:	0x1 - 0xFFFF	Any 16 bits value



## TimerTriggerArm

Address: section "Timer" base address + 0x0008

Description:

Describes what start the timer.

31	30	29	28	27	26	25	24
Soft_TimerArm	Reserved				TimerTriggerOverlap(1:0)		TimerArmEnable
23	22	21	20	19	18	17	16
TimerArmSource(4:0)					TimerArmActivation(2:0)		
15	14	13	12	11	10	9	8
Soft_TimerTrigger	TimerMeasurement	Reserved	TimerTriggerLogicESel(1:0)		TimerTriggerLogicDSel(1:0)		TimerTriggerSource(5)
7	6	5	4	3	2	1	0
TimerTriggerSource(4:0)					TimerTriggerActivation(2:0)		

<b>Soft_TimerArm</b> <i>WO/AutoClr</i>							
	This bit is used to generate a software trigger when the TimerTriggerSource is set to Software. It has no effect otherwise.						
Possible Values:	0x0	Nothing					
	0x1	Software Timer Arm					

<b>TimerTriggerOverlap (1:0)</b> <i>RW</i>							
	<p>This register configs the behaviour of the triggers received when the timer is active(delay or active phases).</p> <p>This feature applies when :</p> <p>1) TimerArmEnable = '0' (ARM bypass)</p> <p>2) TimerTriggerSource != '0' (continuous)</p> <p>-M_OFF: No trigger overlap allowed during the Timer Active period (M_DEFAULT).</p> <p>-M_LATCH: latch any trigger received during the Timer Active period.</p> <p>-M_RESET: Reset the Timer and restart counting if a trigger is received during the Active period</p>						
Value at Reset:	0x0						
Possible Values:	0x0	M_OFF					
	0x1	M_LATCH					
	0x2	M_RESET					
	0x3	Reserved					

<b>TimerArmEnable</b> <i>RW</i>							
	This rregister is the Timer Arm Enable. When set to '1' the timer will wait for an ARM event as defined in register TimerArmSource. When set to 0 the timer will not wait for an ARM event.						
Value at Reset:	0x0						
Possible Values:	0x0	The timer will not wait for a ARM event					
	0x1	The timer will wait for a ARM event					

<b>TimerArmSource (4:0)</b> <i>RW</i>		
	This is the Timer Arm Source selector. When set to off, the timer doesn't wait for a trigger. When select Software as source, select corresponding activation register to rising edge (0x0).	
Value at Reset:	0x0	
Possible Values:	0x0	Software
	0x1 - 0x4	Input Line
	0x5 - 0x7	Internal Line
	0x8 - 0xF	Timer Output

<b>TimerArmActivation (2:0)</b> <i>RW</i>		
	What behavior of the timer arm signal.	
Value at Reset:	0x0	
Possible Values:	0x0	RisingEdge
	0x1	FallingEdge
	0x2	AnyEdge
	0x3	LevelLow
	0x4	LevelHigh
	0x5	None (edge detection disable)

<b>Soft_TimerTrigger</b> <i>WO/AutoClr</i>		
	This bit is used to generate a software trigger when the TimerTriggerSource is set to Software. It has no effect otherwise.	
Possible Values:	0x0	Nothing
	0x1	Software Timer trigger

<b>TimerMeasurement</b> <i>RW</i>		
	<p>Set this field to '1' to use the timer as a pulse width meter. When use the timer as a pulse width meter (or pulse measurement), program the TimerTriggerSource register to the input of the pulse that will be measured. Set the TimerTriggerActivation register to the level we want to measure (low or hi). Set registers TimerClockSource and TimerClockActivation to select the clock used by the counter.</p> <p>To exit from the pulse measurement state, set this register to 0 (back to the WaitOnArm state), or disable the timer (back to TimerDisabled state). When exit the pulse measurement state , the internal counter will be reseted.</p> <p>When the timer is in pulse measurement mode, the output of the timer will be '0'.</p> <p>When the timer is in pulse measurement mode, no need to program registers: DelayClockActivation, DelayClockSource, TimerDelayValue and TimerDuration</p> <p>The maximum pulse width that can be measured with internal default clock(with single edge detection clock activation, rising or falling) is <math>2^{**32} \times 240\text{ns} = 17.17986 \text{ minutes}</math></p>	
Value at Reset:	0x0	
Possible Values:	0x0	Set the timer in pulse generation mode
	0x1	Set the timer in pulse measurement mode

<b>TimerTriggerLogicESel (1:0)</b>		
<i>RW</i>	Logic FlipFlop Enable input Selection	
Value at Reset:	0x0	
Possible Values:	0x0	Logic 1
	0x1	Arm Activation signal
	0x2	Trigger Activation signal AND Arm Activation signal
	0x3	Trigger Activation signal OR Arm Activation signal

<b>TimerTriggerLogicDSel (1:0)</b>		
<i>RW</i>	Logic FlipFlop D input Selection	
Value at Reset:	0x0	
Possible Values:	0x0	Trigger Activation signal
	0x1	Trigger Activation signal AND Arm Activation signal
	0x2	Trigger Activation signal OR Arm Activation signal
	0x3	Trigger Activation signal XOR Arm Activation signal

<b>TimerTriggerSource (5:0)</b>		
<i>RW</i>	<p>This is the Timer trigger Source selector .</p> <p>When the source is set to continuous, the state machine will wait for an ARM(if enabled) and then it will not wait for a trigger. After the first cycle is accomplished, the timer will bypass the wait on arm and wait on trigger states. IDLE-&gt;ARM-&gt;TRIG-&gt;DELAY-&gt;ACTIVE -&gt; DELAY -&gt;ACTIVE -&gt; DELAY -&gt;ACTIVE...</p> <p>When the source is set to other than continuous mode, the state machine will always return to the wait for arm state(single event).</p> <p>When select Software or QuadratureDecoder X Output as source, select corresponding activation register to rising edge (0x0).</p>	
Value at Reset:	0x0	
Possible Values:	0x0	Continuous mode (Delaying->Active->Delaying...)
	0x1	Software
	0x2 - 0x5	Input Line
	0x6 - 0x8	Internal input
	0x9 - 0xC	Tick Table Output
	0xD	QuadratureDecoder Output
	0xE - 0x15	Timer Output

<b>TimerTriggerActivation (2:0)</b>		
<i>RW</i>	What behavior of the timer trigger signal.	
Value at Reset:	0x0	
Possible Values:	0x0	RisingEdge
	0x1	FallingEdge
	0x2	AnyEdge
	0x3	LevelLow
	0x4	LevelHigh
	0x5	None (edge detection disable)

## TimerClockSource

Address: section "Timer" base address + 0x000C

Description:

This is the signal used to clock the timer.

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

<b>IntClock_sel (1:0)</b>	
<i>RW</i>	
This register selects the Clock Int frequency used in the logic.  Default GPM: Tclk = 8*30ns = 240 ns, 4.166666 mhz Default GPM-Atom: Tclk = 8*40ns = 320 ns, 3.125 mhz Default Spider PCIe: Tclk=8*16ns=128 ns, 7.8125 Mhz  If the TimerClockActivation is set to AnyEdge, then the frequency doubles, since the logic is clocked DDR. The default frequency is then 8.333333 mhz for GPM and 6.250mhz for the GPM-Atom.	
Value at Reset:	0x1
Possible Values:	0x0      Clock Int twice the nominal frequency
	0x1      Clock Int is nominal clock period as defined by TimerClockPeriod register
	0x2      Clock Int is half the nominal frequency
	0x3      Clock Int is quarter of the nominal frequency

<b>DelayClockActivation (1:0)</b>	
<i>RW</i>	
What behavior of the delay clock activation signal.	
Value at Reset:	0x0
Possible Values:	0x0      RisingEdge
	0x1      FallingEdge
	0x2      AnyEdge
	0x3      None (edge detection disable)

<b>DelayClockSource (3:0)</b>	
<i>RW</i>	
This is the clock source for the delay phase of the timer.  When select Internal reference clock or QuadratureDecoder X Output as source, select corresponding activation register to rising edge (0x0).	
Value at Reset:	0x0
Possible Values:	0x0      Internal reference clock as defined by Period_ns field
	0x1 - 0x4      Input Line
	0x5      QuadratureDecoder Outputs

<b>TimerClockActivation (1:0)</b>		
<i>RW</i>		
What behavior of the timer clock activation signal.		
Value at Reset:	0x0	
Possible Values:	0x0	RisingEdge
	0x1	FallingEdge
	0x2	AnyEdge
	0x3	None (edge detection disable)

<b>TimerClockSource (3:0)</b>		
<i>RW</i>		
This is the clock source for the main (active) phase of the timer.		
When select Internal reference clock or QuadratureDecoder X Output as source, select corresponding activation register to rising edge (0x0).		
Value at Reset:	0x0	
Possible Values:	0x0	Internal reference clock as defined by Period_ns field
	0x1 - 0x4	Input Line
	0x5	QuadratureDecoder Outputs

# TimerDelayValue

Address: section "Timer" base address + 0x0010

Description:  
length of time the trigger will be delayed before it activates the timer.

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

TimerDelayValue (31:0)		
RW	This is the number of delay clock that a timer is delayed before it start counting in active phase.	
Value at Reset:	0x0	
Possible Values:	0x0 - 0xFFFFFFFF	Any 32 bits value

## TimerDuration

---

Address: section "Timer" base address + 0x0014

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

<b>TimerDuration (31:0)</b>		
<i>RW</i>	This is the number of clock that a timer counts in active phase.	
Value at Reset:	0x1	
Possible Values:	0x0 - 0xFFFFFFFF	Any 32 bits value

# TimerLatchedValue

Address: section "Timer" base address + 0x0018

Description:

This is the current timer value latched when the TimerLatchValue bit is asserted.

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

TimerLatchedValue (31:0)		
RO	This is the Timer Latched value	
Possible Values:	0x0 - 0xFFFFFFFF	Any 32 bits value



## TimerStatus

Address: section "Timer" base address + 0x001C

31	30	29	28	27	26	25	24
TimerStatus(2:0)			TimerStatus_Latched(2:0)			Reserved	
23	22	21	20	19	18	17	16
Reserved						TimerEndIntmaskn	TimerStartIntmaskn
15	14	13	12	11	10	9	8
Reserved					TimerLatchAndReset	TimerLatchValue	TimerCntReset
7	6	5	4	3	2	1	0
Reserved						TimerInversion	TimerEnable

<b>TimerStatus (2:0)</b>		
<i>RO</i>		Returns the current state of the timer
Possible Values:	0x0	TimerDisabled
	0x1	WaitOnArm
	0x2	WaitOnTrigger
	0x3	Delaying, output of the is '0'
	0x4	Active, output of the timer is '1'
	0x5	Measure, output of the is '0'

<b>TimerStatus_Latched (2:0)</b>		
<i>RO</i>		Returns the state of the timer at the time the register TimerLatchValue was set.
		This field may be used with TimerLatchedValue register to allow the driver to know the state of the timer and the value of timer counter.
Possible Values:	0x0	TimerDisabled
	0x1	WaitOnArm
	0x2	WaitOnTrigger
	0x3	Delaying, output of the is '0'
	0x4	Active, output of the timer is '1'
	0x5	Measure, output of the is '0'

<b>TimerEndIntmaskn</b>		
<i>RW</i>		Timer End Interrupt IRQ MASK not. When set to 0, the timer IRQ will be not generated.
Value at Reset:	0x0	
Possible Values:	0x0	No interrupt will be generated for the corresponding End Timer.
	0x1	Interrupt will be generated for the corresponding End Timer.

<b>TimerStartIntmaskn</b>		
<i>RW</i>		Timer Start Interrupt IRQ MASK not. When set to 0, the timer IRQ will be not generated.
Value at Reset:	0x0	
Possible Values:	0x0	No interrupt will be generated for the corresponding Timer.
	0x1	Interrupt will be generated for the corresponding Timer.

<b>TimerLatchAndReset</b> <i>RW</i>		
	When the user enable this field, the internal timer counter will be reseted to '0' when the TimerLatchValue register is toggle or at the end of one pulse mesuement.	
Value at Reset:	0x0	
Possible Values:	0x0	Don't reset the internal counter after latching the current value
	0x1	Reset the internal counter after latching the current value

<b>TimerLatchValue</b> <i>WO/AutoClr</i>		
	Writing 1 in this field will copy the current timer Value into the TimerLatchedValue register. It also copy the current state of the Timer to the field TimerStatus_Latched. This mechanism must be used because the software cannot read the current timer value in a single clock.	
Possible Values:	0x0	Nothing
	0x1	Timer latched value register snapshot

<b>TimerCntrReset</b> <i>WO/AutoClr</i>		
	This is the Timer Counter Reset . When set to '1', the counter in the timer logic will be reset to '0'. The state machine will remain at the current state. This reset is generally used when the timer is set to work as a counter of events.	
Possible Values:	0x0	Nothing
	0x1	Timer Counter Reset

<b>TimerInversion</b> <i>RW</i>		
	Output of the timer can be inverted through this bit.	
Value at Reset:	0x0	
Possible Values:	0x0	Output is '1' in the Active state, '0' otherwise
	0x1	Outout is '0' in the Active state, '1' otherwise

<b>TimerEnable</b> <i>RW</i>		
	This register is the Timer Enable.  When set to 0, the counter is reset to 0, the state machine is set in TimerDisable State(IDLE), and the output signal of the timer is set to 0.	
Value at Reset:	0x0	
Possible Values:	0x0	Timer is disabled, it's output is in '0' state
	0x1	Timer is enabled and cycles in arm, trigger, delay and active state

## Section: Microblaze

Address Range: [0x0A00 - 0x0A7C]

Description:

This section is associated with Microblaze subsystem

### CAPABILITIES\_MICRO

Address: section "Microblaze" base address + 0x0000

Description:

This register identifies the capabilities of Profinet Acceleration Microblaze (AKA Profiblaze).

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

<b>MICRO_ID (7:0)</b>	
<i>STATIC</i>	Any feature with 8 MSB set to 0x70 is a the Microblaze interface as defined in the following registers.
Value at Reset:	0x70

<b>FEATURE_REV (3:0)</b>	
<i>STATIC</i>	Revision of the feature. This field must be used by software to detect if the current software support the register definition of this feature.
Value at Reset:	0x0

<b>Intnum (4:0)</b>	
<i>STATIC</i>	This is the bit number in the interrupt field when the interrupts from this Microblaze are forwarded.
Value at Reset:	0x6

## ProdCons (1:0)

Address: section "Microblaze" base address + 0x0004 + (index \* 0x4)

Description:

Parameters for the Producer-Consumer exchange area

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

<b>MemorySize (4:0)</b> <i>STATIC</i>	
	<p>This is the power of 2 of the size of dual port memory area, defined in bytes.</p> <p>For example, a value of 12 in this field means that there are <math>2^{12}</math> bytes = 4096 bytes for the dual port area. Each regions, assuming the configuration channel is bi-directionnal, will then be 2048 bytes long.</p> <p>The dual-port memory is aligned on its size and placed after the registers. This also means that the actual dual port memory is located 4096 bytes after the value of Offset register.</p>
	Value at Reset: 0xc

<b>Offset (19:0)</b> <i>RO</i>	
	<p>Offset from Spider memory space where the producer-consumer area is located. This is the offset of the first pointer registers. The address of the dual-port memory must be deducted from this offset and the size of the memory.</p> <p>Note that the layout of the producer-consumer register area is defined for the current FEATURE_REV revision. Future implementation with different Pointers size (16 bits for example) would use a different FEATURE_REV.</p>

## Section: AnalogOutput

Address Range: [0x0A80 - 0x0AFC]

Description:

This section is associated with the analog outputs of the design. The FPGA does not have analog output per se, but the external circuitry converts the FPGA output, a PWM signal, into an analog value.

### CAPABILITIES\_ANA\_OUT

Address: section "AnalogOutput" base address + 0x0000

Description:

This register identifies the capabilities of the analog output module of this feature.

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

<b>ANA_OUT_ID (7:0)</b>	
<i>STATIC</i>	Any feature with 8 MSB set to 0x67 is an analog output as defined in the following registers.
Value at Reset:	0x67

<b>FEATURE_REV (3:0)</b>	
<i>STATIC</i>	Revision of the feature. This field must be used by software to detect if the current software support the register definition of this feature.
Value at Reset:	0x0

<b>NB_OUTPUTS (3:0)</b>	
<i>STATIC</i>	This is the number of outputs controlled by this feature, which is also the number of register following this capabilities register.
Value at Reset:	0x1

# OutputValue

Address: section "AnalogOutput" base address + 0x0004

Description:  
This is the output value

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
OutputVal(7:0)							

OutputVal (7:0) RW	
	<p>This register is used to set the almost static output voltage.</p> <p>This is a pre-qualification estimate implementation. It is subject to change. The voltage on the analog output supply should be 13.6V. The count written in this register is the output voltage in 0.1V increment. If a value of 0 is written in the register, the output will be 0 V. If a value of 136 (or more) is written in this register, the ouput should be the full 13.6V.</p>
Value at Reset:	0x0

# Section: EOFM

Address Range: [0x0B00 - 0x0B00]

Description:  
End Of Feature Marker

## EOFM

Address: section "EOFM" base address + 0x0000

Description:  
End Of Feature Marker

31	30	29	28	27	26	25	24
EOFM(7:0)							
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							

EOFM (7:0)	
STATIC	End Of Feature Marker
Value at Reset:	0x0

## External: ProdCons (1:0)

Address Range: [0x2000 - 0x3FFC]

External repeated 2 times. ProdCons(i) base address located @  $0x2000 + (i * 0x2000)$

Description:

Producer-Consumer exchange area.

## Pointers

Address: external "ProdCons" base address + 0x0000

Description:

In this section, INPUT refers to data going from the Host to the Microblaze, Output is for data going from the Microblaze to the Host. Free\_start and Free\_end refer to the section of BRAM that do not contain valid data. It is free as seen from the producer of data.

Note that pointers are defined as 8 bit fields aligned on 8-bit boundaries to be atomically accessible from a LPC bus.

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

<b>OUTPUT_FREE_END (7:0)</b>	
<i>RW</i>	Pointer to the last data processed by the Host. Data up to this location (included) is available to the producer. This field is writable by the Host, read-only on the Microblaze side.
Value at Reset:	0xff

<b>OUTPUT_FREE_START (7:0)</b>	
<i>RO</i>	Pointer where the Microblaze writes data for the Host. The Host must read the data up to this location (not included). This field is writable by the Microblaze, read-only on host side.

<b>INPUT_FREE_END (7:0)</b>	
<i>RO</i>	Pointer to the last data processed by the Microblaze. Data up to this location (included) is available to the producer. This field is writable by the Microblaze, read-only on the Host side.



<b>INPUT_FREE_START (7:0)</b>	
<i>RW</i>	Pointer where the host writes data for the Microblaze. The Microblaze must read input data up to this location (not included). This field is writable by the Host, read-only on Microblaze side.
Value at Reset:	0x0

# DPRAM (1023:0)

Address: external "ProdCons" base address + 0x1000 + (index \* 0x4)

Description:

Dual port ram used to exchange information between the HOST through the LPC and the Processor through the AXI BUS

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

<b>data (31:0)</b> <i>RW</i>	
	Data content is defined by the software. This field is only present to suppress warnings in the generation toolchain.
Value at Reset:	N/A (Non-resettable flip-flops used)