SPEC FMC-TDC developer's manual

Support document for the SPEC TDC driver developers.
Description of the register mapping and the establishment of basic communication.

E. Gousiou | June 2014

A. TDC CHARACTERISTICS

Table 1 lists the characteristics of the FMC TDC board. In this TDC application we are only interested in time difference between rising edges of pulses. Note also that pulses of width <100 ns are considered as noise and are rejected. Typical measurements of this application are 10 ns - 500 us (time difference between pulses).

The ACAM chip is registering all the arriving edges, rising and falling, and the subtraction and pulse rejection takes place at the software level.

Input channels	5 channels TTL with software selectable 50 Ohm termination. Inputs need to be protected against +15V pulses with pulse width > 10us at 50Hz.
Channels enable	Software controlled switch that enables/ disables all 5 channels
Timestamps buffer	Circular buffer that keeps the last 128 pulses (256 rising and falling edges); programmable interrupts implemented based on the number of accumulated timestamps or the amount of elapsed time.
Timestamps precision	+/- 700 ps deviation
Timebase accuracy	+/- 4 ppm from a local TCXO on the FMC board; much better accuracy would be reached when used on a White Rabbit enabled FMC carrier.
Max input pulse rate	31.25 MHz from all 5 channels. If the input rate overpasses this value the user is notified with an interrupt.
Timestamps	Timestamps apply to both rising and falling edges of incoming pulses; on the software level the falling edges are only used for the calculation of the pulse width so that pulses < 100 ns are ignored; rising edges are subtracted between them.
Min input pulse width	100 ns, narrower pulses are ignored on software level by subtracting a falling edge from the previous rising one.
ACAM mode	I-mode, 81ps resolution, +/- 500ps precision (6σ)
Connectors	LEMO 00
FMC connector	Low Pin Count
PCB	6 layers

Table 1: TDC specifications

B. SPEC TDC

The SPEC board can house one FMC TDC mezzanine board. The TDC gateware is running in the Xilinx FPGA on the SPEC carrier. The communication with the PCIe interface takes place through the GN4124-PCIe-bridge chip on the carrier board and the GN4124 core that is instantiated in the TDC gateware.
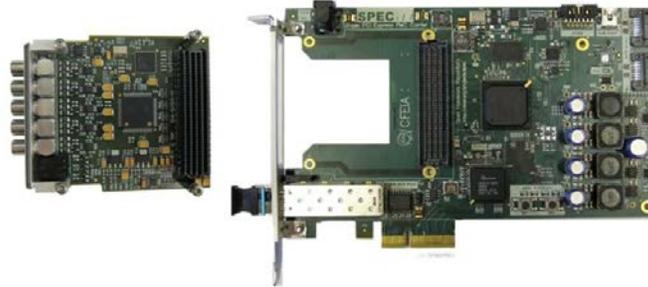

Figure 1: TDC mezzanine and SPEC carrier

C. DATA FORMAT

The TDC gateware is retrieving timestamps generated by the ACAM chip, it is adapting them to a comprehensive format and it is then making them available to the PCIe interface in a circular buffer. Each final timestamp is a 128-bit word with the following structure:

Bits	Description
[127:96]	Metadata [127..125]: Input Channel from 0 to 4 [123] : Edge Type | "1" means rising edge | "0" means falling [122..96] : not used
[95:64]	Local UTC time: the resolution is 1 s
[63:32]	Coarse time within the current UTC time: the resolution is 8 ns
[31:0]	Fine time: the resolution is 81.03 ps

Table 2: Timestamp format. Timestamp [ps] = (Local UTC * 10^{12}) + (Coarse time * 8 * 10^3) + (Fine time * 81.03)

As the structure indicates, each timestamp is referred to a UTC second. The coarse and fine times indicate with 81.03 ps resolution the amount of time passed after the last UTC second.

D. ADDRESS MAPPING

The SPEC TDC gateware is a modular design including the components described in Table 3. The communication with all the components takes place through GN4124 BAR 0; Table 3 lists the base addressing of the different components.

Component	Base Address
Crossbar SDB records	0x00000
Carrier 1-wire (only on the gateware without White Rabbit)	0x10000
Carrier info	0x20000
VIC	0x30000
Mezzanine 1-wire	0x50000
TDC configuration	0x51000
TDC EIC	0x52000
Mezzanine EEPROM I2C	0x53000
TDC timestamps circular buffer	0x54000
White Rabbit core (only on the gateware with White Rabbit)	0x80000

Table 3: Components addressing

Note that the GN4124 chip registers are accessed through BAR 4.

Hereon follows a description of each component and a suggested operation procedure.

I. SDB crossbar

The WISHBONE crossbar implements SDB records [1]. The records describe the WISHBONE slaves and their mapping on the bus. The SDB records ROM is located at offset 0x0. In order to identify the gateware the following SDB meta-information records are used: integration, repo-url, synthesis tool.

II. Carrier 1-wire

Consult [2] for a detailed description of the 1-wire core register map.

Note that in the design with White Rabbit, the carrier 1-wire core is not instantiated, as the White Rabbit core has direct access to the 1-wire line.

III. Carrier info

The following registers are used:

Register name	R/W	Description		Byte Address
Carrier type and PCB version	R	[bits 3..0]	Binary coded PCB layout version	0x20000
		[bits 31..16]	Carrier type : 0x1 = SPEC 0x2 = SVEC 0x3 = VFC 0x4 = SPEXI	
		[bits 15..4]	not used	
Status	R	[bit 0]	FMC presence, active low	0x20004
		[bit 1]	GN4124 clock status, active high	
		[bit 2]	Mezzanine PLL status, active high	
		[bit 3]	Xilinx 62.5 MHz PLL status,only for White Rabbit version	
		[bits 31..3]	not used	

Table 4: Carrier info register map

IV. VIC

In order to redirect interrupts from different cores to the corresponding driver in the Linux kernel in a generic way, a two layers scheme is used. The first layer is the Embedded Interrupt Controllers, described in section VII, which is multiplexing the different interrupt sources into one single line. The second layer is the Vectored Interrupt Controller, VIC, which is multiplexing interrupt lines from different EICs into a single line to the host. In this case, the VIC interrupt request output is connected to the GPIO 8 of the GN4124 chip. The GN4124 must be configured to generate a MSI when a rising edge is detected on GPIO 8. The VIC keeps a table (IVT_RAM) initialized with the base addresses of the EICs connected to each one of its inputs. Here there is only the TDC EIC at address 0x52000. Table 5 describes all the VIC registers.

Register name	R/W			Description	Byte Address
CTL VIC Control reg	R/W	[bit 0]	ENABLE	Write '1' to enable the VIC operation Write '0' to disable the VIC operation	0x30000
		[bit 1]	POL	Write '1' to set the IRQ output active high Write '0' to set the IRQ output active low	
		[bit 2]	EMU_ EDGE	Write '1' to force a low pulse of EMU_LEN--clock-cycles at each write to EOIR. Write '0' for a normal IQR master line behavior	
		[bit 3]	EMU_ LEN	Length of the delay between write to EOIR and re-assertion of irq_master_o	
RISR Raw Irq Status	R	[bit 0] [bits 31..1]		Current state of the FMC TDC mezzanine interrupt Not used	0x30004
IER Irq Enable Reg	W	[bit 0] [bits 31..1]		Write '1' to enable the FMC TDC mezzanine interrupt Write '0' has no effect Not used	0x30008
IDR Irq Disable Reg	W	[bit 0] [bits 31..1]		Write '1' to disable the FMC TDC mezzanine interrupt Write '0' has no effect Not used	0x3000C
IMR Irq Mask	R	[bit 0] [bits 31..1]		Read '1' means FMC TDC mezzanine interrupt is enabled Read '0' means FMC TDC mezzanine interrupt is disabled Not used	0x30010
IMR Vector Address reg	R	[bits 31..0]		Address of the pending interrupt vector, read from the IVT_RAM	0x30014
SWIR Software Irq Reg	W	[bits 31..0]		Writing '1' to one of the bits causes a software emulation of the respective interrupt.	0x30018
EOIR End Of Irq Ack Reg	W	[bit 0] [bits 31..1]		Writing '1' acknowledges an FMC TDC mezzanine pending interrupt Not used	0x3001C
MEM RAM with Interrupt Vector Table	R	[bits 31..0]	IVT_RAM	Contain the address "0x52000" of the TDC EIC WISHBONE slave	0x30020
				Rest of the RAM not used	0x30024..3F

Table 5: VIC register map

As the GN4124 chip is an edge-only IRQ controller, an edge sensitive configuration should be used by the drivers, with the following characteristics:

VIC control register
ENABLE = '1'
POL = '1'
EMU_EDGE = '1'
EMU_LEN = 750
IER = "0x1"

Table 6: VIC configuration for SPEC

V.   Mezzanine 1-wire

Similarly to the Carrier 1-wire of section, consult [2] for a detailed description of the 1-wire core register map.

VI.   TDC configuration

To operate the FMC TDC board it is necessary to give values to certain configuration registers. This includes registers for the configuration of the ACAM chip and other local registers for the TDC core.

TDC configuration: ACAM chip Registers

The following registers are part of the TDC core and are used for the communication with the ACAM chip. For a detailed description of the registers please consult the TDC-GPX documentation [3].

Name	R/W	Description	Address	Typical Value
Acam config reg. 0	R/W	rising/falling edges config	0x51000	x01F0FC81
Acam config reg. 1	R/W	Channel adjustments (other modes)	0x51004	x00000000
Acam config reg. 2	R/W	mode I and disable unused channels according to the application	0x51008	x00000E02
Acam config reg. 3	R/W	resolutions and tests (other modes)	0x5100C	x00000000
Acam config reg. 4	R/W	start timer set to 16 and resets	0x51010	x0200000F
Acam config reg. 5	R/W	start retrigger OFF and offset set to 2.000	0x51014	x000007D0
Acam config reg. 6	R/W	LF flags levels to be defined according to the application	0x51018	x00000003
Acam config reg. 7	R/W	PLL values: RefClkDiv=7, HSDiv=234, PhaseNeg	0x5101C	x00001FEA
Acam config reg. 11	R/W	ERR flag config on the 8 Hit FIFOs	0x5102C	x00FF0000
Acam config reg. 12	R/W	INT flag config on Start nb overflow + HFIFO & IFIFO status flags	0x51030	x04000000
Acam config reg. 14	R/W	16-bit mode control	0x51038	x00000000

Table 7: ACAM configuration registers

Acam config reg. 0: Is set to enable the internal oscillator and the rising and falling edges for the TTL inputs 1 to 5.

Acam config reg. 1: Not used in the ACAM mode chosen for this application.

Acam config reg. 2: Sets the operational mode of the ACAM chip to the I-mode. Disables channels 6 to 8.

Acam config reg. 3: Not used in the ACAM mode chosen for this application.

Acam config reg. 4: Sets the StartTimer to 16; i.e. 512 ns. Sets the EF pin to drive all the time.

Acam config reg. 5: Sets start retrigger to OFF. Sets the programmable internal start offset to 2000.

Acam config reg. 6: Sets the threshold level for the LF flags arbitrary to 3. Can be changed if required for further developments of the application.

Acam config reg. 7: Sets the ACAM internal PLL values. RefClkDiv=7, HSDiv=234 and inverts the phase output.

Acam config reg. 11: Sets the ErrFlag pin to report for any full flags on the HitFIFOs.

Acam config reg. 12: Sets the IntFlag to the highest bit of the Start# (Start number) counter.

TDC configuration: ACAM read-back Registers

A different set of registers is used to store the values of the registers that are read-back directly from the ACAM chip. This set of registers includes the configuration registers detailed above, plus the Read-only registers to access the Interface FIFOs registers as well as the Start01 register.

Name	R/W	Description	Address	Typical Value
Acam readback reg. 0	R	rising/falling edges config	0x51040	xc1F0FC81
Acam readback reg. 1	R	Channel adjustments (other modes)	0x51044	xc0000000
Acam readback reg. 2	R	mode I and disable unused channels	0x51048	xc0000E02
Acam readback reg. 3	R	resolutions and tests (other modes)	0x5104C	xc0000000
Acam readback reg. 4	R	start timer set to 16 and resets	0x51050	xc200000F
Acam readback reg. 5	R	start retrigger OFF and offset set to 2.000	0x51054	xc00007D0
Acam readback reg. 6	R	LF flags levels to max	0x51058	xc00000FC
Acam readback reg. 7	R	PLL values: RefClkDiv=7, HSDiv=234, PhaseNeg	0x5105C	xc0001FEA
Acam readback reg. 8	R	IFIFO 1	0x51060	
Acam readback reg. 9	R	IFIFO 2	0x51064	
Acam readback reg. 10	R	Start01	0x51068	
Acam readback reg. 11	R	ERR flag config on the 8 Hit FIFOs	0x5106C	xc0FF0000
Acam readback reg. 12	R	INT flag config on Start nb overflow + HFIFO & IFIFO status flags	0x51070	xc4000800
Acam readback reg. 14	R	16-bit mode control	0x51078	xc0000000

Table 8: ACAM read-back registers

The following table lists the local configuration registers and the value they should be set to through PCIe writes. Note that the default reset value of the registers is 0x0, apart from the IRQ tstamp thresh (reset value: 0xFF), IRQ time thresh (reset value: 0xC8) and DAC word (reset value: 0xA8F5).

Name	R/W		Description	Byte Address	Typical Set Value
Starting UTC time	R/W	[bits 31..0]	updated on demand by PCI-e or reset	0x51080	
Inputs enable	R/W	[bit 0]	input ch 1 termination enable	0x51084	0x0000009F
		[bit 1]	input ch 2 termination enable		
		[bit 2]	input ch 3 termination enable		
		[bit 3]	input ch 4 termination enable		
		[bit 4]	input ch 5 termination enable		
		[bit 7]	general enable for all channels		
		[bits 5,6,31..8]	not used		
IRQ tstamp thresh	R/W	[bits 7..0]	an interrupt is issued if the number of accumulated timestamps since the last irq exceeds this threshold	0x51090	0x000000FF = full mem
		[bits 31..8]	not used		
IRQ time thresh	R/W	[bits 31..0]	an interrupt is issued if this amount of ms has passed after the last irq and at least a timestamp has been registered	0x51094	0x000000C8 = 200 ms
DAC word	R/W	[bits 23..0]	word to be sent to the DAC	0x51098	0x0000A8F5 = 1.65 V
		[bits 31..24]	not used		
Deactivate channel	R/W	[bit 0]	Deactivate timestamps from ch 1	0x5109C	0x00000000 = all ch active
		[bit 1]	Deactivate timestamps from ch 2		
		[bit 2]	Deactivate timestamps from ch 3		
		[bit 3]	Deactivate timestamps from ch 4		
		[bit 4]	Deactivate timestamps from ch 5		
Current UTC time	R	[bits 31..0]	calculated by the core according to the local 125 MHz clk and the "staring utc time" register	0x510A0	
Circular buffer write pointer	R	[bits 11..0]	number of 8-bit-words to be read from the circular buffer = number of 128-bit-timestamps*16	0x510A8	
Da Capo counter		[bits 31..12]	number of times the circular buffer has been overwritten		
White Rabbit ctrl	R/W	[bit 0]	Enables White Rabbit	0x510B0	
White Rabbit status	R	[bit 1]	high if White Rabbit is synthesized	0x510B4	
	R	[bit 2]	tm_link_up		
	R	[bit 4]	tm_aux_clk_locked		
	R	[bit 5]	tm_time_valid		
	R	[bit 9]	tm_ aux_clk_lock_en		
Control Register	W	[bits 11..0]	Commands the main core FSM	0x510FC	See Table 10
		[bits 31..12]	Not used		

Table 9: TDC core local registers

Amongst the registers for the operation of the TDC core, one in particular is utterly important: the Control Register allows commanding the main Finite State Machine.

Control Register Bit	Action Description	Control Register Value
Bit 0	Activate acquisition	x00000001
Bit 1	De-activate acquisition	x00000002
Bit 2	Load ACAM config	x00000004
Bit 3	Read ACAM configuration	x00000008
Bit 4	Read ACAM status	x00000010
Bit 5	Read ACAM IFIFO 1	x00000020
Bit 6	Read ACAM IFIFO 2	x00000040
Bit 7	Read ACAMStart01 register	x00000080
Bit 8	Reset ACAM chip	x00000100
Bit 9	Load UTC time	x00000200
Bit 10	Clear Da Capo flag	x00000400
Bit 11	Configure DAC by sending the DAC word	x00000800

Table 10: Control register actions

Read/Write

Starting UTC time: Sets the initial value for the TDC core internal time base to which all timestamps will be referenced. Note that since in this application we are only interested in differences between timestamps, the actual UTC time is not significant (it is eliminated in the subtraction).

Input enable controls: Controls the terminations on each input as well as the general enable of the inputs.

DAC word: Word to be sent to the TDC mezzanine DAC. Note that the control register has to be activated for the reconfiguration of the DAC to take place (11th bit). Note also that the reconfiguration of the DAC is always followed by the reconfiguration of the local PLL. The default value sets the DAC to its middle value.

IRQ timestamps threshold: Sets the threshold according to which interrupts on IRQ register bit 0 are issued. If the accumulated timestamps after the last IRQ (or the beginning of time) exceed this threshold then an interrupt is raised. The default value is 256 timestamps, which is the full memory.

IRQ time threshold: Sets the threshold according to which interrupts on IRQ register bit 1 are issued. If the amount of ms that have passed since the last IRQ (or the beginning of time) exceeds this threshold and at least one timestamp has been registered, then an interrupt is raised. The default value is 0xC8 that is 200 ms.

Deactivate channel: Each of the last 5 bits of the register is used for the deactivation of a channel. Note that the deactivation takes place inside the TDC core and not at the ACAM level; the ACAM continues giving all the timestamps and if a channel is deactivated (through the corresponding register bit), the received timestamps are not being registered in the circular buffer.

White Rabbit control: Bit 0 enables the startup of White Rabbit synchronization.

Current UTC time: As the TDC core keeps track of UTC time according to its local oscillator, this registers provides the current local value used for the timestamps, in order for the software application to perform the correspondent correction with respect to the official UTC. Whenever the drift between the local UTC and the official UTC needs to be corrected, the new value for the local UTC is set and updated through the corresponding command of the Control Register. It is not necessary to stop the acquisition for this.

WR pointer: Keeps track of the next position to be written in the circular buffer memory for the timestamps (12 LSb). It includes the 'Da Capo counter' that keeps track of the number of overruns of the memory block (20 MSb).

White Rabbit status: White Rabbit status register carries information of different White Rabbit signals. In particular [bit 4] shows if the synchronization has been established.

Write only

Control register: Only one bit at a time can be activated since each bit carries a command. The value is cleared upon writing.

VII.   TDC EIC

The TDC EIC gathers the interrupts from the TDC core. The TDC core can generate an interrupt in any of the following three cases:

- o when the amount of timestamps written in the "circular_buffer", since the last interrupt or since the startup of the acquisition, exceeds the PCIe settable threshold irq_tstamp_threshold. We refer to this interrupt as "timestamps interrupt".
- o when some timestamps have been written in the circular_buffer (>=1 timestamp) and the amount of time passed since the last interrupt or since the acquisition startup, exceeds the PCIe settable threshold irq_time_threshold. We refer to this interrupt as "time interrupt".
- o when the ACAM raises the Error flag; this means that the ACAM Hit FIFOs have been receiving pulses with a frequency > 31.25 MHz. We refer to this interrupt as "acam error interrupt".

The three inputs are multiplexed in the EIC and the result is forwarded to the VIC. Interrupt sources can be masked using the enable and disable registers. Table 11 describes the TDC EIC registers.

Name	R/W	Description		Byte Address
EIC IRR Interrupt disable register	W	Writing '1' disables the handling of the interrupt associated with the corresponding bit. Writing '0' has no effect.		0x52000
		[bit 0]	write '1' to disable "tstamps irq"	
		[bit 1]	write '1' to disable "time irq"	
		[bit 2]	write '1' to disable "acam error irq"	
		[bits 31..3]	not used	
EIC IER Interrupt enable register	W	Writing '1' enables the handling of the interrupt associated with the corresponding bit. Writing '0' has no effect.		0x52004
		[bit 0]	write '1' to enable "tstamps irq"	
		[bit 1]	write '1' to enable "time irq"	
		[bit 2]	write '1' to enable "acam error irq"	
		[bits 31..3]	not used	
EIC IMR Interrupt mask register	R	Shows which interrupts are enabled. Reading '1' means that the interrupt associated with the bitfield is enabled.		0x52008
		[bit 0]	read '1' means "tstamps irq" is enabled	
		[bit 1]	read '1' means "time irq" is enabled	
		[bit 2]	read '1' means "acam error irq" is enabled	
		[bits 31..3]	not used	
EIC ISR Interrupt status register	R/W	Each bit represents the state of the corresponding interrupt. Reading '1' means the interrupt is pending. Writing '1' to a bit clears the corresponding interrupt. Writing '0' has no effect.		0x5200C
		[bit 0]	read '1' means "tstamps irq" is pending read '0' means no pending interrupt write '1' to clear the "tstamps irq" write '0' has no effect	
		[bit 1]	read '1' means "time irq" is pending read '0' means no pending interrupt write '1' to clear the "time irq" write '0' has no effect	
		[bit 2]	read '1' means "acam error irq" is pending read '0' means no pending interrupt write '1' to clear the "t acam error irq" write '0' has no effect	
		[bits 31..3]	not used	

Table 11: TDC EIC registers

VIII.   Mezzanine EEPROM I2C

Consult [4] for a detailed description of the I2C EEPROM core register map.

IX.   TDC timestamps circular buffer

The timestamps that are retrieved from the ACAM and are formatted in the 128-bit words of Table 2 are finally stored in a circular buffer accessible through base address 0x54000. The "write pointer" described in Table 9 indicates how many bytes (= timestamps*16) are available for reading.

Name	R/W	Byte Address
timestamp #0	R	0x54000
timestamp #1	R	0x54016
…	…	…
timestamp #256	R	0x54FFF

Table 12: Circular buffer map

X.   LEDs

There are 6 orange LEDs on the front panel of the TDC board.

LED STA blinks upon a "local PPS pulse".

LED#1 blinks upon the writing in the "circular buffer" of a timestamp referred to channel 1. If the input termination for channel 1 is ON, there is a blinking when the timestamp is written; if the input termination is OFF, the LED is always ON and it turns OFF when the timestamp is written.

LED#2..LED#5 function accordingly for channels 2..5. The signals regarding the channel number and the writing in the "circular buffer" come from the "data formatting" unit.
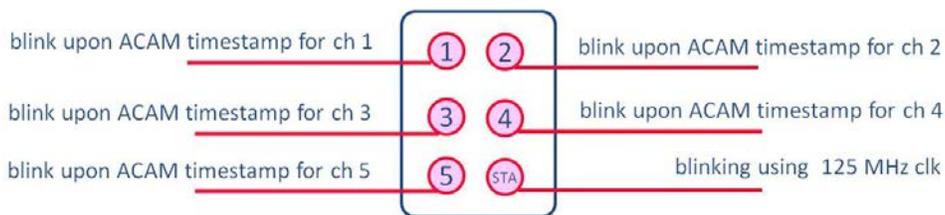

Figure 2: TDC mezzanine board front panel LEDs

E. CONFIGURATION SEQUENCE

The following points describe the steps that need to be followed on the software level so as to operate the TDC board right after powering it up.

I. At power-up

- o Load the carrier FPGA with the TDC core **bitstream.**

- o Set the clock frequency for the **GNUM** chip is set to 200 MHZ through reg. 4:808.

- o Force a **reset** on the RST_N output pin of the GNUM (reg. 4:800). This launches the initialization sequence of the FPGA that sets the parameters for the PLL on the TDC mezzanine. This configuration lasts for ~1ms and after that the mezzanine 1-wire and I2C can be accessed*.

- o After a reset, the **FSM** of the core is in the "**inactive**" state, which means that the configuration registers can be accessed. Write all the configuration registers described in Table 7 into the TDC core.

- o Load the ACAM configuration registers **to the ACAM chip**. The command to load them is issued through enabling the control register bit 2 (see Table 10).

- o Optionally, for **verification**, read back the configuration of the ACAM by enabling the control register bit 3 (see Table 10) and reading the corresponding Read-back Registers (see Table 8).

- o **Reset the ACAM** chip through the control register bit 8 (see Table 10).

- o Optionally read back for verification the **Status Register of the ACAM**, by enabling the control register bit 4 (see Table 10) and reading the corresponding Read-back Register (see Table 8).

- o Configure the **VIC** (see Table 6) and enable the **TDC EIC** (see Table 11).

- o Optionally, configure the **DAC** by writing a new DAC word on the corresponding TDC core local register (see Table 9) and enabling the control register bit 11; the default value is 1.65 V, in the middle of the range.

- o Optionally, for TDC#1 and TDC#2 configure the **interrupt thresholds** (Table 9) by writing on the corresponding TDC core local registers (see Table 9); the default value for the tstamps_irq is 256 (circular buffer full) and for the time_irq 200 ms.

- o The reference starting time for the **local UTC** can be set through the corresponding TDC core register (see Table 9) and loaded through the control register bit 9 (Table 10).

- o Enable the **inputs** and the desired termination resistors though the dedicated TDC core local register (Table 9).

- o Optionally, enable White Rabbit through the White Rabbit control register bit 0; wait until the synchronization is established by checking the White Rabbit status register bit 4 (see Table 9).

- o **Launch** the **acquisition** through the control register bit 1 (see Table 10). This generates the **TStart** signal for the ACAM chip, and from that moment on, every pulse arriving to the ACAM inputs will generate a timestamp that will be immediately fetched by the TDC core and stored in the circular buffer memory.

* At this point the reading of the mezzanine EEPROM should take place for the calibration of the board. This document however does not include the calibration details. Consult [4] for all the calibration information.

The following points describe the steps that need to be followed on the software level so as to operate the TDC board right after TDC core reset.

II. After a reset of the TDC core

- After a reset, the **FSM** of the core is in the "**inactive**" state, which means that the configuration registers can be accessed. Write all the configuration registers described in Table 7 into the TDC core.

- Load the ACAM configuration registers **to the ACAM chip**. The command to load them is issued through enabling the control register bit 2 (see Table 10).

- Optionally, for **verification**, read back the configuration of the ACAM by enabling the control register bit 3 (see Table 10) and reading the corresponding Read-back Registers (see Table 8).

- **Reset the ACAM** chip through the control register bit 8 (see Table 10).

- Optionally read back for verification the **Status Register of the ACAM**, by enabling the control register bit 4 (see Table 10) and reading the corresponding Read-back Register (see Table 8).

- Configure the **VIC** (see Table 6) and enable the **TDC EIC** (see Table 11).

- Optionally, configure the **DAC** by writing a new DAC word on the corresponding TDC core local register (see Table 9) and enabling the control register bit 11; the default value is 1.65 V, in the middle of the range.

- Optionally, for TDC#1 and TDC#2 configure the **interrupt thresholds** (Table 9) by writing on the corresponding TDC core local registers (see Table 9); the default value for the tstamps_irq is 256 (circular buffer full) and for the time_irq 200 ms.

- The reference starting time for the **local UTC** can be set through the corresponding TDC core register (see Table 9) and loaded through the control register bit 9 (Table 10).

- Enable the **inputs** and the desired termination resistors though the dedicated TDC core local register (Table 9).

- Optionally, enable White Rabbit through the White Rabbit control register bit 0; wait until the synchronization is established by checking the White Rabbit status register bit 4 (see Table 9).

- **Launch** the **acquisition** through the control register bit 1 (see Table 10). This generates the **TStart** signal for the ACAM chip, and from that moment on, every pulse arriving to the ACAM inputs will generate a timestamp that will be immediately fetched by the TDC core and stored in the circular buffer memory.

III.   Operation

o   The software should be in mode of **expecting interrupts**.

o   When an interrupt arrives, the "write pointer" register (see Table 9) should be read so as to know how many timestamps are available in the **circular buffer**.

o   The timestamps are retrieved by reading this amount of bytes starting from address 0x54000.

o   The interrupt should then be cleared and the driver should go back to the mode of waiting for a new interrupt.

Note that in the responsibilities of the driver is to discard pulses narrower than 100 ns. Every rising edge timestamp of a channel should be subtracted by the following falling edge (of the same channel) so as to confirm this pulse length.

F. References

[1]: SDB records: http://www.ohwr.org/projects/fpga-config-space/wiki

[2]: Open cores One Wire master: http://opencores.org/project,sockit_owm

[3]: Open cores I2C master: http://opencores.org/project,sockit_owm

[4]: FMC TDC gateware guide: http://www.ohwr.org/projects/fmc-tdc-1ns-5cha-gw/repository/revisions/master/show/doc