FMC-TDC developer’s manual

*Support document for the TDC drivers developers*

E. Gousiou | April 2013

# 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 : TDC specifications

# ADDRESS MAPPING

To operate the FMC TDC board it is necessary to give values to certain configuration registers.

All the registers are accessed through the GNUM’s Base Address Register spaces (BAR) and are described in the SDB space.

* The GNUM chip registers are accessed through BAR 4
* All other registers are mapped to BAR 0. The table below lists the base addressing of the different components of the TDC gateware design

|  |  |
| --- | --- |
| Component | Base Address |
| GNUM core DMA configuration | 0x4000 |
| Carrier 1-wire | 0x4800 |
| Carrier CSR information | 0x4C00 |
| TDC core | 0x5000 |
| Interrupts controller | 0x5400 |
| Mezzanine EEPROM I2C | 0x5800 |
| Mezzanine 1-wire | 0x5C00 |

Table : Components addressing

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

### GNUM core Registers:

The following registers belong to the GNUM core and their use is explained in detail in the [corresponding documentation](http://www.ohwr.org/projects/gn4124-core/wiki).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | R/W | description | Address | Typical Value |
| DMACTRLR | R/W | DMA engine control | 0:4000 | x00000000 |
| DMASTATR | R | DMA engine status | 0:4004 | x00000001 |
| DMACSTARTR | R/W | DMA start address in the carrier | 0:4008 | Last read address |
| DMAHSTARTLR | R/W | DMA start address (low) in the PCIe host | 0:400C | From Page List |
| DMAHSTARTHR | R/W | DMA start address (high) in the PCIe host | 0:4010 | x00000000 |
| DMALENR | R/W | DMA read length in bytes | 0:4014 | Last timestamp address i.e. last address read |
| DMANEXTLR | R/W | Pointer (low) to next item in list | 0:4018 | x00000000 |
| DMANEXTHR | R/W | Pointer (high) to next item in list | 0:401C | x00000000 |
| DMAATTRIBR | R/W | DMA chain control | 0:4020 | x00000000 |

Table : GNUM core addressing

### TDC core: 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](http://www.physics.utoronto.ca/~astummer/pub/mirror/Projects/2011_PhotonFinish2__Octal_Coincidence_Timer/Docs/Acam%20GPX%20time-to-digital%20converter.pdf).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | R/W | Description | Address | Typical Value |
| Acam config reg. 0 | R/W | rising/falling edges config | 0:5000 | x01F0FC81 |
| Acam config reg. 1 | R/W | Channel adjustments (other modes) | 0:5004 | x00000000 |
| Acam config reg. 2 | R/W | mode I and disable unused channels according to the application | 0:5008 | x00000E02 |
| Acam config reg. 3 | R/W | resolutions and tests (other modes) | 0:500C | x00000000 |
| Acam config reg. 4 | R/W | start timer set to 16 and resets | 0:5010 | x0200000F |
| Acam config reg. 5 | R/W | start retrigger OFF and offset set to 2.000 | 0:5014 | x000007D0 |
| Acam config reg. 6 | R/W | LF flags levels to be defined according to the application | 0:5018 | x00000003 |
| Acam config reg. 7 | R/W | PLL values: RefClkDiv=7, HSDiv=234, PhaseNeg | 0:501C | x00001FEA |
| Acam config reg. 11 | R/W | ERR flag config on the 8 Hit FIFOs | 0:502C | x00FF0000 |
| Acam config reg. 12 | R/W | INT flag config on Start nb overflow + HFIFO & IFIFO status flags | 0:5030 | x04000000 |
| Acam config reg. 14 | R/W | 16-bit mode control | 0:5038 | x00000000 |

Table : 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 operational 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 operational 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.

Since all of these ACAM registers are Read/Write, the read-back of their value is stored in the **ACAM ReadBack Registers [0. .14]:** 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 | 0:20040 | xc1F0FC81 |
| Acam readback reg. 1 | R | Channel adjustments (other modes) | 0:20044 | xc0000000 |
| Acam readback reg. 2 | R | mode I and disable unused channels | 0:20048 | xc0000E02 |
| Acam readback reg. 3 | R | resolutions and tests (other modes) | 0:2004C | xc0000000 |
| Acam readback reg. 4 | R | start timer set to 16 and resets | 0:20050 | xc200000F |
| Acam readback reg. 5 | R | start retrigger OFF and offset set to 2.000 | 0:20054 | xc00007D0 |
| Acam readback reg. 6 | R | LF flags levels to max | 0:20058 | xc00000FC |
| Acam readback reg. 7 | R | PLL values: RefClkDiv=7, HSDiv=234, PhaseNeg | 0:2005C | xc0001FEA |
| Acam readback reg. 8 | R | IFIFO 1 | 0:20060 |  |
| Acam readback reg. 9 | R | IFIFO 2 | 0:20064 |  |
| Acam readback reg. 10 | R | Start01 | 0:20068 |  |
| Acam readback reg. 11 | R | ERR flag config on the 8 Hit FIFOs | 0:2006C | xc0FF0000 |
| Acam readback reg. 12 | R | INT flag config on Start nb overflow + HFIFO & IFIFO status flags | 0:20070 | xc4000800 |
| Acam readback reg. 14 | R | 16-bit mode control | 0:20078 | xc0000000 |

Table : ACAM read-back registers

### TDC core Registers: core configuration

The following registers are used for the configuration of the TDC core.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | R/W | Description | Address | Typical Value |
| Starting UTC time | R/W | is updated on demand by a PCI-e command or reset | 0:5080 |  |
| Inputs enable | R/W | controls the termination enabling for each input (bits 4 downto 0) and general enable input (bit 7) | 0:5084 | x0000009F |
| IRQ tstamp thresh | R/W | an interrupt is issued if the number of accumulated timestamps since the last irq exceeds this threshold (only 7 LSbits considered) | 0:5090 | x000000FF  = memory full |
| IRQ time thresh | R/W | an interrupt is issued if this amount of seconds has passed after the last irq and at least a timestamp has been registered | 0:5094 | x00000078  = 2 min |
| DAC word | R/W | word to be sent to the TDC mezzanine DAC (only 24 LSbits considered) | 0:5098 | x0000A8F5  = 1.65 V |
|  |  | RESERVED | 0:509C |  |
| Current UTC time | R | calculated by the core according to the local clk | 0:50A0 |  |
| Circular buffer write pointer | R | provided to PCI-e for DMA configuration  (includes the DaCapo flag) | 0:50A8 |  |
| Control Register | W | Commands the main core state machine | 0:50FC |  |

Table : TDC core 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 Acam Start01 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 | x00000400 |

Table : 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 2 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 3 are issued. If the amount of seconds that have passed since the last IRQ (or the beginning of time) exceed this threshold and at least one timestamp has been registered, then an interrupt is raised. The default value is 0x78 sec, that is 2 min.

#### Read only

**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).

#### 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.

### Interrupts Controller:

Each bit of the Interrupts register represents a different type of interrupt:

|  |  |
| --- | --- |
| IRQ register bit | Description |
| Bit 0 | GNUM core DMA |
| Bit 1 | GNUM core DMA |
| Bit 2 | TDC core threshold on number of accumulated timestamps |
| Bit 3 | TDC core threshold on elapsed time since the last interrupt |
| Bit 4 | ACAM error flag |
| Bits 5..31 | not used |

### Carrier CSR:

The following registers are used:

|  |  |  |  |
| --- | --- | --- | --- |
| Register | Address | Bits | Description |
| Reg. 0 | 0x4C00 | [3..0] | PCB version |
| Reg. 0 | 0x4C00 | [4] | FMC PLL status, active high |
| Reg. 0 | 0x4C00 | [31..16] | Carrier type |
| Reg. 1 | 0x4C04 | [0] | FMC presence, active low |
| Reg. 1 | 0x4C04 | [1] | CNUM4124 status, active high |

# 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.

### At power-up:

* Load the carrier FPGA with the TDC core [**bitstream**](http://www.ohwr.org/projects/fmc-tdc/repository/changes/firmware/tdc.bit)**.**
* Set the clock frequency for the **GNUM** chip is set to 200 MHZ through reg. 4:808.
* 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 local PLL on the TDC mezzanine.
* After a reset, the **FSM** of the core is in the “**Acquisition inactive**” state, which means that the configuration registers can be accessed. Write all the configuration registers into the TDC core.
* Load the ACAM configuration registers **to the ACAM chip**. The command to load them is issued through the corresponding bit (bit 2) of the Control Register.
* Optionally, for **verification**, read back the configuration of the ACAM by issuing the corresponding command through the Control Register bit 3 and reading the corresponding Read-back Registers.
* **Reset the ACAM** chip through the Control Register bit 8.
* Optionally read back for verification the **Status Register of the ACAM**, by issuing the corresponding command through the Control Register bit 4 and reading the corresponding Read-back Register.
* Enable the **interrupts** controller; confirm that the thresholds for the timestamps and time interrupts are set.
* Optionally, configure the **DAC** by writing a new DAC word on the corresponding TDC core register and setting the Control Register bit 11; the default value is 1.65 V, in the middle of the range.
* The reference starting time for the **local UTC** can be set through the corresponding TDC core register and loaded for operation with a command on the Control Register bit 9.
* Enable the **inputs** and the desired termination resistors though the dedicated TDC core register.
* **Launch** the **acquisition** through the corresponding command of the Control Register (bit 1). 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.

### 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.

### After a reset of the TDC core

* A After a reset, the **FSM** of the core is in the “**Acquisition inactive**” state, which means that the configuration registers can be accessed. Write all the configuration registers into the TDC core.
* Load the ACAM configuration registers **to the ACAM chip**. The command to load them is issued through the corresponding bit (bit 2) of the Control Register
* Optionally, for **verification**, read back the configuration of the ACAM by issuing the corresponding command through the Control Register bit 3 and reading the corresponding Read-back Registers.
* **Reset the ACAM** chip through the Control Register bit 8.
* Optionally read back for verification the **Status Register of the ACAM**, by issuing the corresponding command through the Control Register bit 4 and reading the corresponding Read-back Register.
* Enable the **interrupts** controller; confirm that the thresholds for the timestamps and time interrupts are set.
* Optionally, configure the **DAC** by writing a new DAC word on the corresponding TDC core register and setting the Control Register bit 11; the default value is 1.65 V, in the middle of the range.
* The reference starting time for the **local UTC** can be set through the corresponding TDC core register and loaded for operation with a command on the Control Register bit 9.
* Enable the **inputs** and the desired termination resistors though the dedicated TDC core register.
* **Launch** the **acquisition** through the corresponding command of the Control Register (bit 1). 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.

### Operation:

The software could stay in mode of **expecting interrupts**. When an interrupt arrives, the circular buffer WR Register is read so as to know how many timestamps are available in the **Circular Buffer**.

Then a **DMA** can be performed accordingly. In order to configure the DMA, at least the DMACSTARTR, DMAHSTARTLR and DMALENR registers in the GNUM core need to be set. Then the DMA is launched through the DMACTRLR and its success can be verified in the DMASTATR.

# DATA FORMAT

Each timestamp is 128 bits. It therefore appears in the host memory in 4 consecutive 32-bit words:

* [127..96] : Metadata
  + [127..125] : Input Channel
  + [123] : Edge Type; “1” means rising edge, “0” means falling
  + [122..96] : used for debugging
* [95..64] : Local UTC with a 1s resolution.
* [63..32] : Coarse time within the second with a 8 ns resolution.
* [31..0] : Fine time to be added to the coarse time. Each bit representing 81 ps.