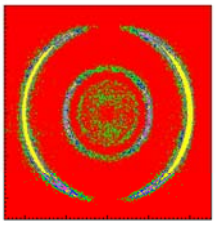
RoentDek
Handels GmbH

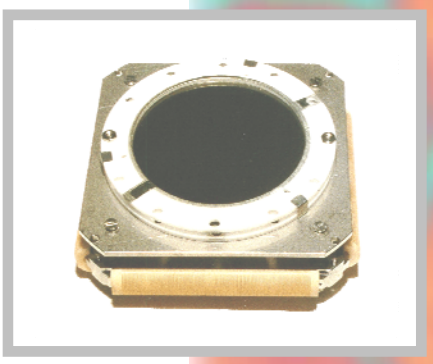
Supersonic Gas Jets
Detection Techniques
Data Acquisition Systems
Multifragment Imaging Systems

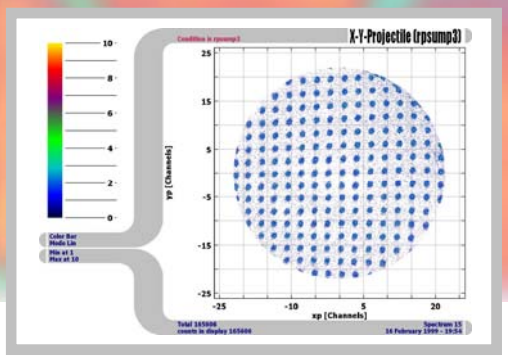


TDC4HM System Manual

High precision TDC for highest signal rates

Version (11.0.1702.1)





Mail Addresses:

Headquarter

RoentDek Handels GmbH
Im Vogelshaag 8
D-65779 Kelkheim-Ruppertshain
Germany

Frankfurt subsidiary

RoentDek Handels GmbH
c/o Institut für Kernphysik
Max-von-Laue Str. 1
D-60438 Frankfurt am Main
Germany

Web-Site:

www.roentdek.com

WEEE:

DE48573152

Table of Contents

1 Introduction

The **TDC4HM** is a *common-start* time-to-digital converter. The timestamps of leading or trailing edges of digital pulses are recorded. The **TDC4HM** produces a stream of output packets, each containing data from a single start event, i.e. the relative timestamps of all stop pulses that occur within the user defined range.

Figure 1.1: TDC4HM PCIe Card

1.1 Features

- 4 channel common start TDC with 13 ps resolution
- Standard Range: 218 µs (24 bit timestamp)
- Extended Range: 13,975 µs
- Bin size: approx. 13 ps
- Double pulse resolution: 5 ns
- Dead time between groups: none
- Maximum start rate: 4 MHz
- L0 FIFO: 15 words/channel
- L1 FIFO: 512 words/channel
- L2 FIFO: 10000 words
- PCIe 1.1 x1 with 200 MB/s throughput

2 Hardware

2.1 Installing the TDC4HM

- Shut down your computer
- For your devices safety, turn off the power to your computer and all peripheral devices.
- Drain static electricity from your body by touching the metal chassis (the unpainted metal at the back of your computer)
- For your personal safety, remove the power cord from your computer
- Remove the cover of the computer as described in your computer's manual.
- Locate a free PCIe x1 slot (or higher amount of lanes) in your computer, and firmly insert the card into the selected slot. To avoid damaging your hardware, insert the card only into a slot with the same bus type as the card. Inserting the card into any other type of slot can damage your card, your computer, or both.
- Firmly secure the adapter with a screw (or clip), to ensure that the adapter is properly grounded to the computer's chassis.
- Replace the cover of the computer as described in your computer's manual.

2.2 TDC4HM Inputs and Connectors

2.2.1 Connectors

The inputs of the **TDC4HM** are located on the PCIe bracket. Figure 2.3 shows the location of the start input S and the four stop inputs A to D. Lemo-00 connectors are used for input connection. The inputs are AC-coupled and have an impedance of 50 Ω - a schematic of the input circuit is shown in Figure 2.2. the digital trigger threshold can be adjusted in order to comply with a manifold of single ended signal standards enabling the acquisition of positive or negative pulses.

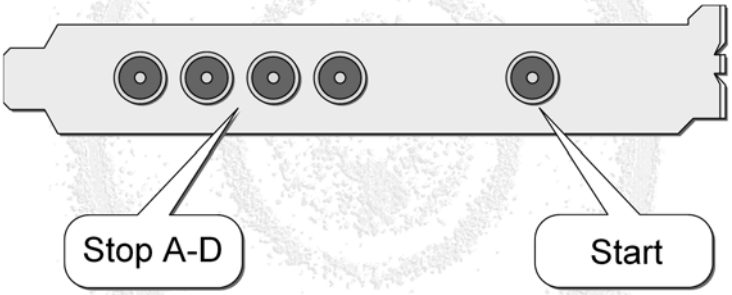


Figure 2.1: Input connectors of the TDC4HB located on the PCIe bracket.

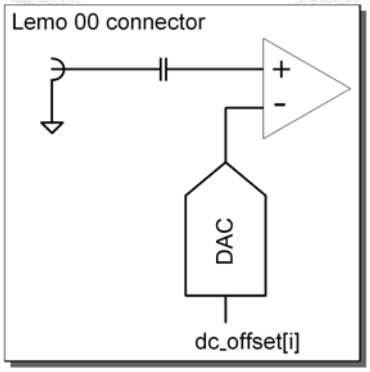


Figure 2.2: Input circuit for each of the five input channels.

Pin	Name
1, 2	GND
3, 4	external CLK in N, external CLK in P
5, 6	GND
7, 8	reserved/NC
9, 10	GND
11, 12	reserved/NC
13, 14	GND
15, 16	reserved/NC
17, 18	GND
19, 20	reserved/NC
21, 22	GND
23, 24	reserved/NC
25, 26	GND
27, 28	reserved/NC
29, 30	GND
31, 32	reserved/NC
33, 34	GND

Table 2.1: Pinout of connector C2

Figure 2.3: Schematic view of a TDC4HM board showing inter-board connectors C1 and C2.

Furthermore, three board interconnection connectors can be found at the top edge of the board, as displayed in Figure 2.3. Connector C1 (labelled J25 on the board) is reserved for future use. The pinout of connector C2 (labelled J12 on the board) is given in Table 2.1 and the pinout of connector C3 (labelled as J6 on the board) is depicted in Table 2.2.

Pin	Name
1	+3.3 V
2 - 9	reserved/NC
10	GND

Table 2.2: Pinout of connector C3.

2.3 TDC4HM Functionality

The TDC4HM is a "classic" *common start* time-to-digital converter. It records the time difference between leading or trailing edges on the start input and the stop inputs. Each stop channel A-D can be enabled individually. The accuracy of the acquired timestamps is approximately 8 ps. The timestamps are recorded in integer multiples of a bin size of 13.02083 ps (76.8 GHz). The data acquisition can be limited to rising or falling signal transitions. Transitions of the input signals are called hits. To reliably detect hits the signal has to be stable for at least 500 ps before and after the edge. Between multiple hits on a stop channel a deadtime of approximately 5 ns occurs. Within this deadtime, further hits on the stop channel are reported with a coarse timestamp only. The maximum trigger rate on the start channel is 4 MHz.

2.3.1 Grouping and Events

In typical applications a start hit is followed by a manifold of hits on e.g. a detector. The hits recorded are managed in groups (which are called in some applications "events"). Figure 2.4 shows a corresponding timing diagram. The user can define the range of a group, i.e. the time window within which hits on the stop channels are recorded, in software. Hits occurring outside of that time window are discarded. The maximum recording range for a group is 218 µs. Different ranges can be set for each of the 4 stop channels by setting corresponding channel.start and channel.stop values in the channel configuration. The values need to be set as multiples of 13.02083 ps. A value of 768 corresponds, for example to a time of 10 ns.

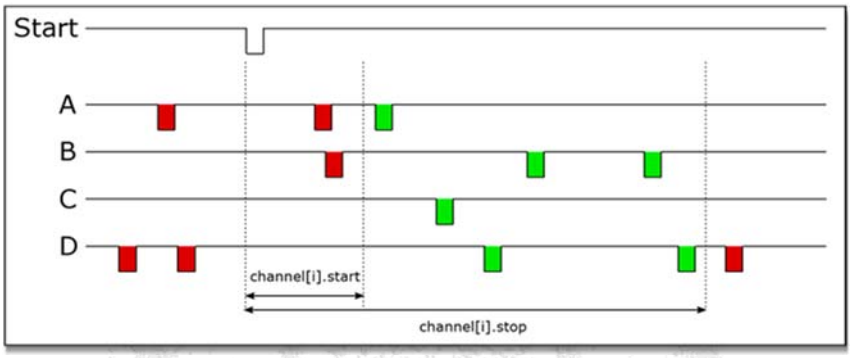


Figure 2.4: Schematic view of a TDC4HM board showing inter-board

2.4 Performing a firmware update

After installing the **TDC4HM** device driver, a firmware update tool is available. By choosing "FirmwareGUI.exe" a firmware update can be performed. After invoking the application a window as shown in Figure 2.5 will appear. The tool can be used for updating the firmware and to create a backup of the on-board calibration data of the TDC4HM unit. If several boards are present, the one which is going to be used can be selected in the upper left corner of the window. Pressing the "Backup" buttons a backup of the firmware or the calibration data will be created, respectively. In order to perform a firmware update, chose the ".cronorom"-file to used by pressing "Browse". The file contains the firmware data. By pressing "Flash" the firmware is written to the board. "Verify" can be used to compare the firmware data stored on the TDC4HM to the one inside a file. "Flash All" and "Verify All" perform the corresponding operation on all boards which are installed.

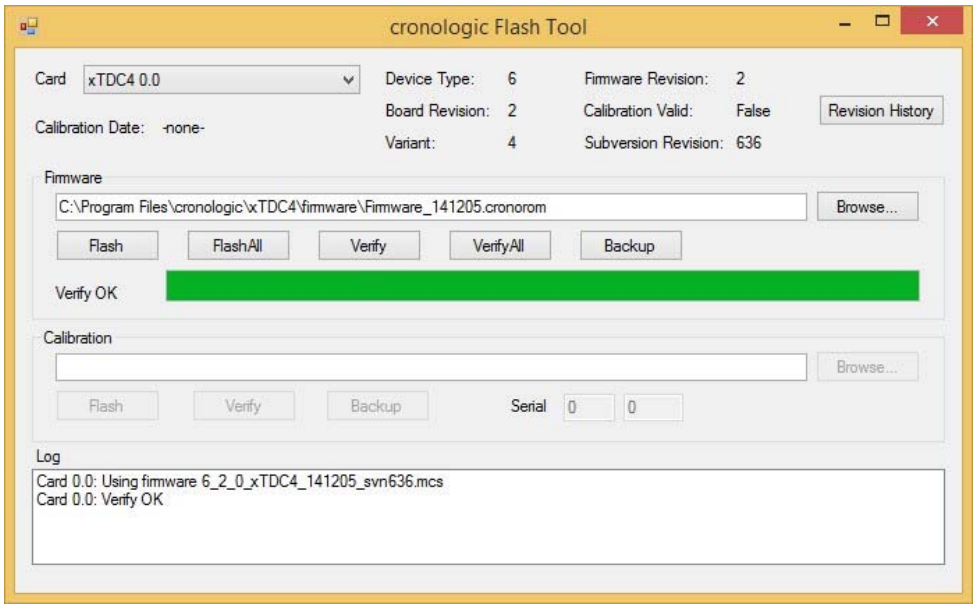


Figure 2.5: The firmware update and calibration data backup tool as provided with the TDC4HM device driver.

Important note: The new firmware will only be used after a power cycle, i.e. after switching the PC (or Ndigo crate) off and back on. A simple reboot is not sufficient. Therefore the information shown in the upper half of the application window does not change right after flashing a new firmware.

2.5 Calibrating the Carry-Chain TDC

After an update of the **TDC4HM** firmware the Carry-Chain TDC may need to be calibrated.
Please perform this procedure only if RoentDek instructed you to do so.
Before calibration make sure to power-cycle the system after updating the TDC4HM firmware. The calibration is done with the tool "XTDC4Calibration.exe" (see Figure 2.6) which is available after installing the **TDC4HM** device driver. Connect an external pulse signal to the Start and channel inputs. The signal should be low active. The pulse low and high width has to be at least 10 ns each. Use "Calibrate" to start the calibration procedure. Follow the on-screen instructions to gather calibration data on all channels. When all channels are calibrated use "Write" to permanently store the calibration data in the **TDC4HM** 4's on-board flash.

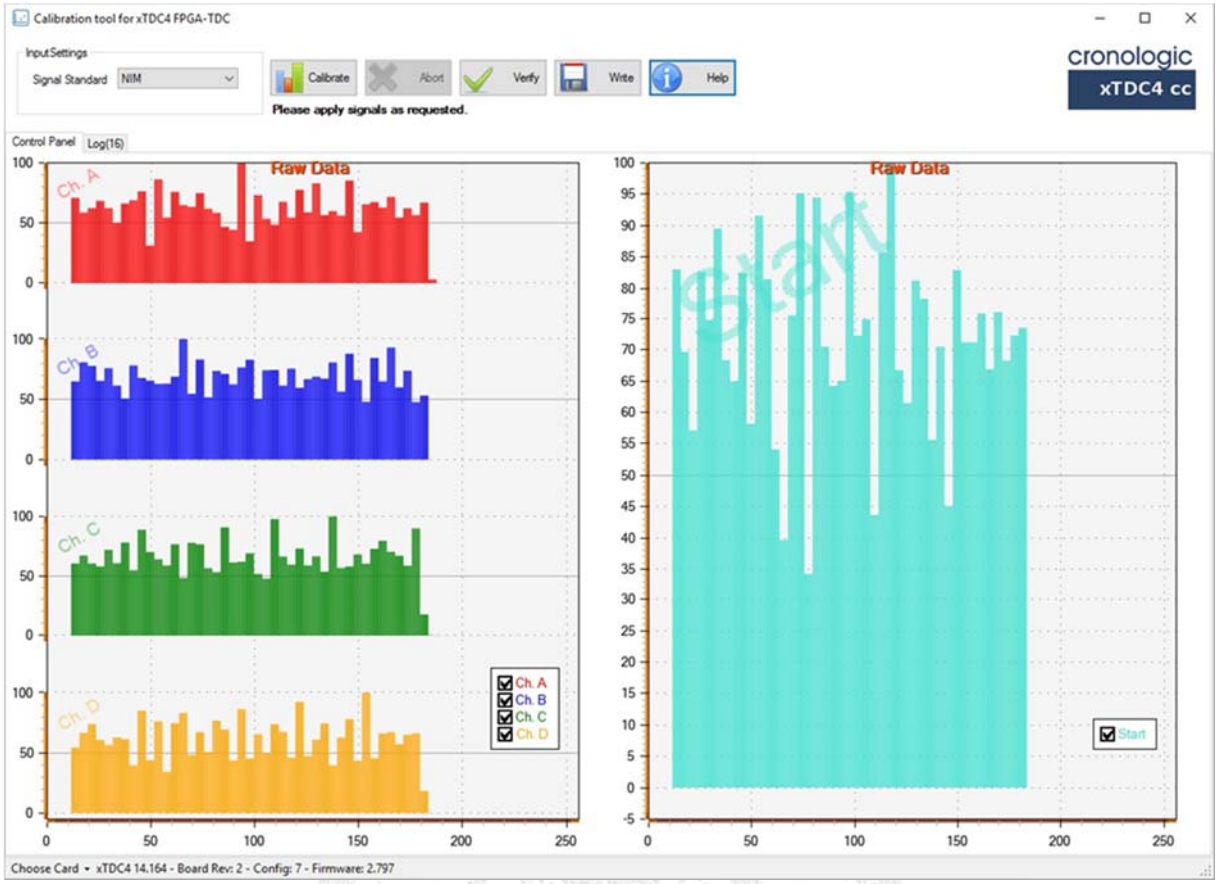


Figure 2.6: The TDC4HM Carry Chain TDC calibration tool.

3 Driver Programming API

The API is a DLL with C linkage. There exists also a .Net wrapper.
The functions provided by the DLL are declared in the file "xTDC4_interface.h".

3.1 Constants

#define xTDC4_CHANNEL_COUNT_4
The number of analog input channels.

#define xTDC4_TIGER_COUNT_5
The number of timing generators.

#define xTDC4_TRIGGER_COUNT_16
The number of triggers. Two per analog input, one per digital input plus some specials.

3.2 Initialization

int xtdc4_close(xtdc4_device *device)
Finalize the driver for this device.

int xtdc4_count_devices(int *error_code, char **error_message)
Return the number of boards that are supported by this driver in the system.

int xtdc4_get_default init parameters(xtdc4_init_parameters *init)
Sets up the standard parameters. Gets a set of default parameters for xtdc4_init(). This must always be used to initialize the xtdc4_init_parameter() structure.

xtdc4_device *xtdc4_init(xtdc4_init_parameters *params, int *error_code, char **error_msg)
Open and initialize the **TDC4HM** board with the given index. With error code and error message the user must provide pointers where to buffers where error information should be written by the driver. Params is a structure of type xtdc4_init_parameters that must be completely initialized.

3.2.1 Structure xtdc4_init_parameters

int version
The version number. Must be set to ***XTDC4_API_VERSION***

int card_index
The index in the list of **TDC4HM** boards that should be initialized.

There might be multiple boards in the system that are handled by this driver as reported by 9

xtdc4_count_devices
This index selects one of them. Boards are enumerated depending on the PCIe slot. The lower the bus number and the lower the slot number the lower the card index.

int board_id
the global index in all cronologic devices.

This 8 bit number is filled into each packet created by the board and is useful if data streams of multiple boards will be merged. If only **TDC4HM** cards are used this number can be set to the card index. If boards of different types that use a compatible data format are used in a system each board should get a unique id. Can be changed with ***int xtdc4_set_board id(xtdc4_device *device, int board_id***.

int64 buffer size[8]
The minimum size of the DMA buffer.
If set to 0 the default size of 16 MByte is used. For the **TDC4HM** only the first entry is used.

int buffer_type
The type of buffer. Can be either allocated (only option currently) or physical.
#define XTDC4_BUFFER_ALLOCATE 0

#define XTDC4_BUFFER_USE_PHYSICAL 1

int64 buffer_address
The start address of the reserved memory.
The buffers will be allocated with the sizes given above. Make sure that the memory is large enough.

int variant
Set to 0. Can be used to activate future device variants such as different base frequencies.

int device_type
A constant for the different devices of ***CRONO DEVICE ****.
Initialized by ***xtdc4_get_default_init_parameters()***. Must be left unchanged.

#define CRONO_DEVICE_HPTDC 0
#define CRONO_DEVICE_NDIGO5G 1
#define CRONO_DEVICE_NDIGO250M 2
#define CRONO_DEVICE_xTDC4 6
#define CRONO_DEVICE_TIMETAGGER4 8

int dma_read_delay
The update delay of the writing pointer after a packet has been send over PCIe. The base unit is 16 to 32 ns. Should not be changed by the user.

int use_ext_clock
If set to 1 use external 10 MHz reference. If set to 0 use internal reference.

3.3 Status Information

3.3.1 Functions for Information Retrieval

The driver provides functions to retrieve detailed information on the type of board, its configuration, settings and state. The information is split according to its scope and the computational requirements to query the information from the board.

int xtdc4_get_fast_info(xtdc4_device *device, xtdc4_fast_info *info)
Returns fast dynamic information.
This call gets a structure that contains dynamic information that can be obtained within a few microseconds.

int xtdc4_get_param_info(xtdc4_device *device, xtdc4_param_info *info)
Returns configuration changes.
Gets a structure that contains information that changes indirectly due to configuration changes.

int xtdc4_get_slow_info(xtdc4_device *device, xtdc4_slow_info *info)
Returns slow dynamic information.
The data reported in this structure requires milliseconds to be obtained. The application should only call it in situation where the program flow can cope with an interruption of that magnitude.

int xtdc4_get_static_info(xtdc4_device *device, xtdc4_static_info *info)
Contains static information.
Gets a structure that contains information about the board that does not change during run time.

3.3.2 Structure xtdc4_static_info

This structure contains information about the board that does not change during run time. It is provided by the function ***xtdc4_get_static_info***.

int size
The number of bytes occupied by the structure.

int version
The version number.

int board_id
ID of the board.
This value is passed to the constructor. It is reflected in the output data.

int driver_revision
Encoded version number.
The lower three bytes contain a triple level hierarchy of version numbers, e.g. 0x010103 encodes version 1.1.3. A change in the first digit generally requires a recompilation of user applications. Change in the second digit denote significant improvements or changes that don't break compatibility and the third digit changes with minor bug fixes and similar updates.

int firmware_revision
Revision number of the FPGA configuration.

int board_revision
Board revision number.
The board revision number can be read from a register. It is a four bit number that changes when the schematic of the board is changed.
0: Experimental first board version. Labeled "Rev. 1"
1: First commercial version. Labeled "Rev. 2"

int board_configuration
Describes the schematic configuration of the board.
The same board schematic can be populated in multiple variants. This is a four bit code that can be read from a register.

int subversion_revision
Subversion revision id of the FPGA configuration.

int chip_id
16 bit factory ID of the TDC chip.

int board_serial
Serial number.
With year and running number in 8.24 format. The number is identical to the one printed on the silvery sticker on the board.

unsigned int flash_serial_high
high 32 bits of 64 bit manufacturer serial number of the flash chip.

unsigned int flash_serial_low
low 32 bits of 64 bit manufacturer serial number of the flash chip

int flash_valid
If not 0 the driver found valid calibration data in the flash on the board and is using it.

3.3.3 Structure xtdc4_param_info

This struct contains configuration changes provided by ***xtdc4_get_param_info()***.

int size
The number of bytes occupied by the structure.

int version
The version number.

double binsize
Bin size (in ps) of the measured TDC data. The TDC main clock is running at a frequency of 76.8 GHz resulting in a bin size of ≈ 13.0208 ps.

int board_id
Board ID.

The board uses this ID to identify itself in the output data stream. The ID takes values between 0 and 255.

int channels
Number of channels in the current ADC mode. Can take values 1, 2 and 4.

int channel_mask
Bit assignment of each enabled input channel.
Mask assigns a certain bit to each enabled input channel.

int64 total_buffer
The total amount of DMA buffer in bytes.

3.3.4 Structure xtdc4 fast info

int size
The number of bytes occupied by the structure.

int version
The version number.

int tdc_rpm
Speed of the TDC fan. Reports 0 if no fan is present.

int fpga_rpm
Speed of the FPGA fan. Reports 0 if no fan is present.

int alerts
Alert bits from temperature sensor and the system monitor.
bit 0: TDC temperature alert (> 141° C)

int pcie_pwr_mgmt
int pcie_link_width
Number of PCIe lanes the card uses.

int pcie_max_payload
Maximum size in bytes for one PCIe transaction. Depends on system configuration.

3.4 Configuration

The device is configured with a configuration structure. The user should first obtain a structure that contains the default settings of the device read from an on board ROM, than modify the structure as needed for the user application and use the result to configure the device.

int xtdc4_configure(xtdc4_device *device, xtdc4_configuration *config)
Configures xtdc4_device.

int xtdc4_get_current_configuration(xtdc4_device *device, xtdc4_configuration *config)
Gets current configuration. Copies the current configuration to the specified config pointer.

int xtdc4_get_default_configuration(xtdc4_device *device, xtdc4_configuration *config)
Gets default configuration. Copies the default configuration to the specified config pointer

3.4.1 Structure xtdc4_configuration

This is the structure containing the configuration information. It is used in conjunction with xtdc4_get_default_***configuration()***, ***xtdc4_get_current_configuration()*** and ***xtdc4_configure()***.
It uses internally the structures ***xtdc4_tiger_block*** and ***xtdc4_trigger***.

int size
The number of bytes occupied by the structure.

int version
A version number that is increased when the definition of the structure is changed. The increment can be larger than one to match driver version numbers or similar. Set to 0 for all versions up to first release.

int tdc_mode
TDC mode. Can be grouped or continuous. Currently supported: grouped.

crono_bool_t start_rising
Rising or falling edge trigger.

double dc_offset[XTDC4_CHANNEL_COUNT + 1]
Set the switching voltage for the input channels S, A - D (see Figure 3.1).

dc_offset[0] : Start
dc_offset[1 - 4] : A – D

Supported range is -1.32 V to +1.18 V. This should be close to 50% of the height of the input pulse. Examples for various signaling standards are defined as follows:

#define DCOFFSET_P_NIM	+0.35
#define DCOFFSET_P_CMOS	+1.18
#define DCOFFSET_P_LVCMOS_33	+1.18
#define DCOFFSET_P_LVCMOS_25	+1.18
#define DCOFFSET_P_LVCMOS_18	+0.90
#define DCOFFSET_P_TTL	+1.18
#define DCOFFSET_P_LVTTL 33	+1.18
#define DCOFFSET_P_LVTTL 25	+1.18
#define DCOFFSET_P_SSTL_3	+1.18
#define DCOFFSET_P_SSTL_2	+1.18
#define DCOFFSET_N_NIM	-0.35
#define DCOFFSET_N_CMOS	1.32
#define DCOFFSET_N_LVCMOS33	-1.32
#define DCOFFSET_N_LVCMOS25	-1.25
#define DCOFFSET_N_LVCMOS18	-0.90
#define DCOFFSET_N_TTL	-1.32
#define DCOFFSET_N_LVTTL_33	-1.32
#define DCOFFSET_N_LVTTL_25	-1.25
#define DCOFFSET_N_SSTL_3	-1.32
#define DCOFFSET_N_SSTL_2	-1.25

The inputs are AC coupled. Thus, the absolute voltage is not important for pulse inputs. It is the relative pulse amplitude that causes the input circuits to switch. ***dc_offset*** must be set to the relative switching voltage for the input standard in use. If the pulses are negative, a negative switching threshold must be set and vice versa.

xtdc4 trigger_trigger[XTDC4 TRIGGER_COUNT]
Configuration of the external trigger sources.

xtdc4_tiger_block tiger_block[XTDC4_TIGER_COUNT]
Configuration of the timing generator.

xtdc4_channel channel[XTDC4_CHANNEL_COUNT]
Configure polaritiy, type and threshold for the TDC channels.

xtdc4_lowres_channel lowres_channel[XTDC4_LOWRES CHANNEL_COUNT]
Not applicable for normal **TDC4HM**, only available for xTDC4-Sciex. Configure polarity, type and threshold for the digital channels.

int auto_trigger_period

int auto_trigger_random_exponent

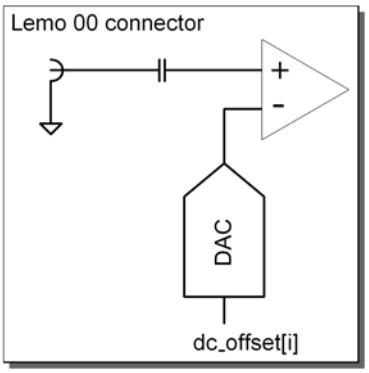


Figure 3.1: Input circuit for each of the five input channels. Both inputs of the buffer are biased at 1.32V by default.

Create a trigger either periodically or randomly. There are two parameters M = trigger period and N = random exponent that result in a distance between triggers of T clock cycles.
T = 1 + M + [1...2N] (3.1)
0 ≤ M < 232 (3.2)
0 ≤ N < 32 (3.3)

There is no enable or reset as the usage of this trigger can be configured in the trigger block channel source field.

Equation 3.1: $\boldsymbol{T = 1 + M + [1..2^N]}$

Equation 3.2: $\boldsymbol{0 \leq M < 2^{32}}$

Equation 3.3: $\boldsymbol{0 \leq N < 32}$

There is no enable or reset as the usage of this trigger can be configured in the trigger block channel source field.

3.4.2 Structure xtdc4_trigger

crono_bool_t falling
Triggers on falling edges.

crono_bool_t rising
Triggers on rising edges.

3.4.3 Structure xtdc4_tiger_block

crono_bool_t enable
Activates timing generator.

crono_bool_t negate
Inverts output polarity. Default is set to false.

crono_bool_t retrigger
Enables/disables retrigger setting.
Default is set to false. If retriggering is enabled the timer is reset to the value of the start parameter, whenever the input signal is set while waiting to reach the stop time.

crono_bool_t extend
Not implemented.

crono_bool_t enable_lemo_output
Enables the LEMO output.

int start
Precursor. Relative to the trigger signal, the data is delayed by the 'start' time interval in units of 6.6 ns (150 MHz). Thus, also data prior to the trigger event is recorded.

int stop
Postcursor.
For edge triggering: this is the total length of the recorded signal.

int sources
A bit mask with a bit set for all trigger sources that can trigger this channel. Default is ***XTDC4_TRIGGER_SOURCES***.

```
#define XTDC4_TRIGGER_SOURCE_S          0x00000001
#define XTDC4_TRIGGER_SOURCE_A          0x00000002
#define XTDC4_TRIGGER_SOURCE_B          0x00000004
#define XTDC4_TRIGGER_SOURCE_C          0x00000008
#define XTDC4_TRIGGER_SOURCE_D          0x00000010
#define XTDC4_TRIGGER_SOURCE_AUTO       0x00004000
#define XTDC4_TRIGGER_SOURCE_ONE        0x00008000
```

3.4.4 Structure xtdc4_channel

Contains TDC channel settings.

crono_bool_t enabled
Enable TDC channel.

crono_bool_t rising
Set whether to record rising or falling edges.

crono_bool_t cc_enable
Enable carry chain TDC. Default is true as initialized by ***xtdc4_get_default_configuration()***. Shall be left unchanged.

crono_bool_t cc_same_edge
Set whether the carry chain TDC records the same edge as THS788 (as backup) or opposite edge. Default is true as initialized by ***xtdc4_get_default_configuration()***. Shall be left unchanged.

crono_bool_t ths788_disable
Disable THS788 timestamps. Default is false as initialized by ***xtdc4_get_default_configuration()***. Shall be left unchanged.

int start
Veto function. Only timestamps posterior to 'start' are recorded.

int stop
Veto function. Only timestamps prior to 'stop' are recorded.

3.5 Run Time Control

int xtdc4 continue_capture(xtdc4_device *device)
Call this to resume data acquisition after a call to ***xtdc4_pause_capture()***.

int xtdc4 pause_capture(xtdc4_device *device)
Pause data acquisition.

int xtdc4 start_capture(xtdc4_device *device)
Start data acquisition.

int xtdc4 start_tiger(xtdc4_device *device)
Start timing generator.

int xtdc4 stop_capture(xtdc4_device *device)
Stop data acquisition.

int xtdc4 stop_tiger(xtdc4_device *device)
Stop timing generator

3.6 Readout

int xtdc4_acknowledge(xtdc4_device *device, crono_packet *packet)
Acknowledges the processing of the last read block. This is only necessary if ***xtdc4_read()*** is not called.
This feature allows to either free up partial DMA space early if there will be no call to ***xtdc4_read*** anytime soon. It also allows to keep data over multiple calls to ***xtdc4_read*** to avoid unnecessary copying of data.

int xtdc4_get_device_type()
Returns the type of the device. Either ***CRONO_DEVICE_XTDC45G*** or ***CRONO_DEVICE_XTDC4250M***

const char* xtdc4_get_last_error message(xtdc4_device *device)
Returns most recent error message.

int xtdc4 read(xtdc4_device *device, xtdc4 read in *in, xtdc4 read out *out)
Return a pointer to an array of captured data in ***read out***. The result can contain any number of packets of type ***xtdc4 packet***. ***read in*** provides parameters to the driver. A call to this method automatically allows the driver to reuse the memory returned in the previous call. Returns an error code as defined in the structure ***xtdc4 read out.***

3.6.1 Input Structure xtdc4_read_in

xtdc4_bool_t acknowledge_last_read
If set ***xtdc4_read()*** automatically acknowledges packets from the last read.

3.6.2 Input Structure xtdc4_read_out

crono packet *first_packet
Pointer to the first packet that was capture by the call of xtdc4 read.

crono packet *last_packet
Address of header of the last packet in the buffer.

int error_code
Assignments of the error codes.

```
#define CRONOREAD_OK                0
#define CRONOREAD_NO_DATA           1
#define CRONOREAD_INTERNAL_ERROR    2
#define CRONOREAD_TIMEOUT           3
```

const char *error_message

3.7 Packet Format

3.7.1 Output Structure crono_packet

unsigned char channel
Unused, always 0.

unsigned char card
Identifies the source card in case there are multiple boards present. Defaults to 0 if no value is assigned to the parameter ***board_id*** in Structure ***ndigo_init_parameters***.

unsigned char type
The data stream consists of 32 bit unsigned data as signified by a value of 6.

unsigned char flags

#define XTDC4_PACKET_FLAG_ODD_HITS 1
The last data word in the data array consists of one timestamp only which is located in the lower 32 bits of the 64 bit data word (little endian).
#define XTDC4_PACKET_FLAG_SLOW_SYNC 2
Start pulse distance is larger than the extended timestamp counter range.

#define XTDC4_PACKET_FLAG_START_MISSED 4
The trigger unit has discarded packets due to a full FIFO.
#define XTDC4_PACKET_FLAG_SHORTENED 8
The trigger unit has shortened the current packet due to full FIFO.
#define XTDC4_PACKET_FLAG_DMA_FIFO_FULL 16
The internal DMA FIFO was full. Might or might not result in dropped packets.
#define XTDC4_PACKET_FLAG_HOST_BUFFER_FULL 32
The host buffer was full. Might or might not result in dropped packets.

unsigned int length
Number of 64-bit elements (each containing up to 2 TDC hits) in the data array.

unsigned int64 timestamp
Coarse timestamp of the start pulse. Values are given in multiples of 1.6 ns.

unsigned int64 data[1]
TDC hits. the user can cast the array to uint32* to directly operate on the TDC hits.

# bits	31 to 8	7 to 4	3 to 0
content	TDC DATA	FLAGS	CHN

Table 3.1 Bit Coding of Hits

The timestamp of the hit is stored in bits 31 down to 8. Bits 7 down to 4 are hit flags:

Bits 7 + 6: Timestamp consists of FPGA timing only with 1666 ps precision, as both the TDC and the Carry Chain TDC missed this stop event.

Bit 7: FPGA did not see the stop event: hit may be out of sequence and may belong to another group.

Bit 6: Timestamp consists of Carry Chain TDC timing only with 150 ps precision, as the TDC missed this hit.

Bit 5: Time since start pulse is longer than timestamp counter range (218µs). If the timestamp counter range is exceeded before the next occurrence of a stop event, a data[] element with bit 5 set which does not belong to a hit on any channel is added to the sequence of hits. The channel number for this hit is set to 15. For each overflow of the timestamp counter in a group one such data[] element is added to the group. The total offset of a hit can be computed by

Equation 3.4: $$\Delta T_{hit} = \#\,\boldsymbol{bits} * \mathbf{218}\,\boldsymbol{\mu s} + \boldsymbol{fine\ timestamp}$$

where # bits counts the number of 5th bit occurrences until the hit shows up in the group. The maximum counter range supported by the **TDC4HM** hardware is 13,975 µs.

Bit 4: Timestamp of the rising edge. Default is falling edge.
The channel number is given in the lowest nibble of the data word. A value of 0 corresponds to channel A, a value of 3 to channel D. Values 7 to 14 are reserved and 15 denotes a group time counter overflow.

4 Technical Data

4.1 TDC Characteristics

Each board is tested against the values listed in the 'Min' column. 'Typical' is the mean value of the first 10 boards produced.

Symbol	Parameter	Min	Typical	Max	Units
INL	Integral nonlinearity				bins
DNL	Differential nonlinearity				bins
t_{Bin}	Binsize		13.02083		ps
t_{Res}	Resolution		8		ps

Table 4.1: TDC Characteristics

4.2 Electrical Characteristics

4.2.1 Oscillator

The **TDC4HM** uses an oscillator with 25 ppb stability.

4.2.2 Environmental Conditions for Storage

The board is designed to be operated under the following conditions:

Symbol	Parameter	Min	Typical	Max	Units
T	ambient temperature	5		40	◦C
RH	relative humidity at 31˚C	20		75	%

Table 4.2: Environmental Conditions for Storage

4.2.3 Environmental Conditions for Storage

The board shall be stored between operation under the following conditions:

Symbol	Parameter	Min	Typical	Max	Units
T	ambient temperature	-30		60	◦C
RH	relative humidity at 31˚C non condensing	10		70	%

Table 4.3: Environmental Conditions for Storage

4.2.4 Power Supply

Symbol	Parameter	Min	Typical	Max	Units
I	PCIe 3,3V rail power consumption			4	mA
VCC	PCIe 3,3V rail power supply	3,1	3,3	3,5	V
I	PCIe 12V rail power consumption			2,1	A
VCC	PCIe 12V rail power supply	11,1	12	12,9	V
I	PCIe 3,3VAux rail power consumption		0		A
VCC	PCIe 3,3VAux rail power supply		3,3		V

Table 4.4: Power Supply

4.2.5 Inputs

The **TDC4HM**-PCIe's inputs are AC-coupled 50 Ω single ended.

Symbol	Parameter	Min	Typical	Max	Units
VBase	Baseline shift				V
tPulse	Pulse length	2	5	200	ns
Z_P	input impedance		50		Ω

Table 4.5: Inputs

4.3 Manufacturer

The **TDC4HM** is a product of:

RoentDek Handels GmbH
Im Vogelshhag 8
65779 Kelkheim / Germany

4.4 Intended Use and System Integration

The **TDC4HM** is designed to comply with DIN EN 61326-1 when operated on a PCIe compliant main board housed in a properly shielded enclosure. When operated in a closed standard compliant PC enclosure the device does not pose any hazards as defined by EN 61010-1.

Radiated emissions, noise immunity and safety highly depend on the quality of the enclosure. It is the responsibility of the system integrator to ensure that the assembled system is compliant to applicable standards of the country that the system is operated in, especially with regards to user safety and electromagnetic interference. Compliance was only tested for attached cables shorter than 3m. All power supplied to the system must be turned off before installing the board. When handling the board, adequate measures have to be taken to protect the circuits against electrostatic discharge (ESD).

4.5 Cooling

The **TDC4HM** in its base configuration has passive cooling that requires a certain amount of air flow. If the case design can't provide enough air flow to the board, a slot cooler like Zalman ZM-SC100 can be placed next to the board. Active cooling is also available as an option to the board.

4.6 Environmental Conditions

The board is designed to be operated under the following conditions:

Symbol	Parameter	Min	Typical	Max	Units
T	ambient temperature	5		40	◦C
RH	relative humidity at 31°C	20		75	%

Table 4.6: Operating Enviromental Conditions

and shall be stored between operation under the following conditions:

Symbol	Parameter	Min	Typical	Max	Units
T	ambient temperature	-30		60	◦C
RH	relative humidity at 31°C non condensing	10		70	%

Table 4.7: Storage Enviromental Conditions

4.7 Inputs

All inputs are AC coupled. The inputs have very high input bandwidth requirements and therefore there are no circuits that provide over voltage protection for these signals. Any voltage on the inputs above 5 V or below -5 V relative to the voltage of the slot cover can result in permanent damage to the board.

4.8 Recycling

RoentDek GmbH is registered with the "Stiftung Elektro-Altgeräte Register" as a manufacturer of electronic systems with Registration ID DE48573152. The **TDc4HM** belongs to category 9, "Überwachungs- und Kontrollinstrumente für ausschliesslich gewerbliche Nutzung". The last owner of a **TDC4HM** must recycle it or treat the board in compliance with §11 and §12 of the German ElektroG or return it to **RoentDek.**

List of Figures

List of Tables

List of Equations