

CDF Note 6999
Version 0.01
June 29, 2004

TDC-II Design and Specification: Run IIB TDC for the COT

Mircea Bogdan, Harold Sanders, Henry Frisch
The University of Chicago

Ting Miao, Steve Chappa, Bob DeMaat
Fermilab

1. Introduction

This note describes the design and implementation of the Chicago TDC-II board.

2. Principle of operation

The principle of the Time to Digital conversion is presented in Figure 1, as seen in the QuartusII simulation window. The incoming LVDS signal is applied to a dedicated input pin (`serdes_in`) of an Altera Stratix FPGA.

This FPGA is provided with a dedicated megafunction, “`altlvds_rx`”, which implements a deserialization LVDS receiver, working on a 1.2 ns input data rate.

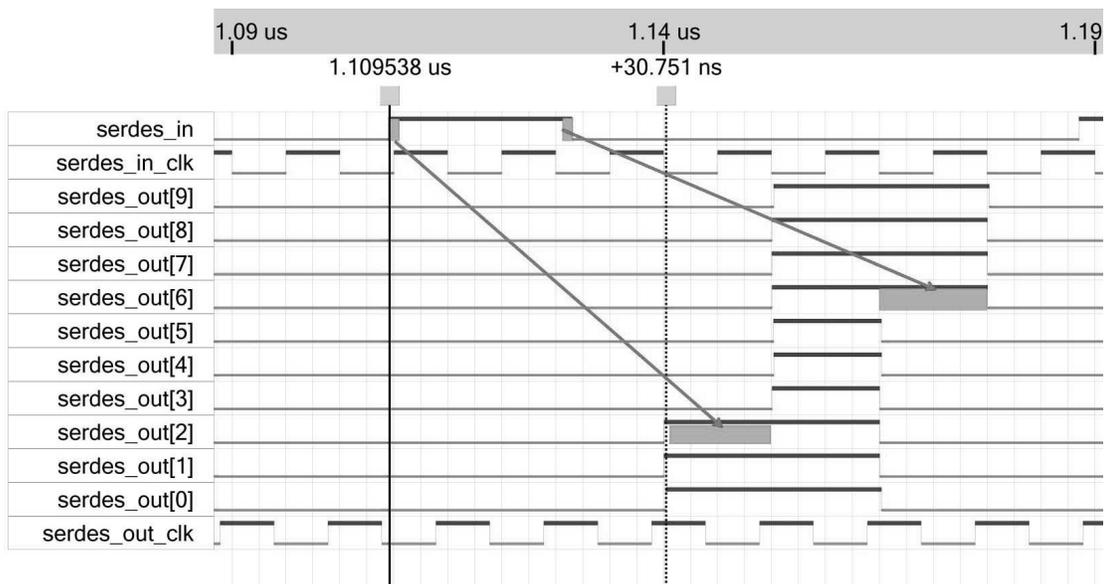


Figure 1. Principle of operation

The input pulse is converted into a 10-bit data stream, moving on a 12ns clock period. The leading edge is determined by counting the number of “0” bits after a L1 Accept pulse and before the first “1” bit. The pulse width is calculated by counting the number of successive “1” bits.

3. TDC Chip - Block diagram

The block diagram of the TDC Chip is presented in Figure 2. Each TDC Chip accepts 48 LVDS inputs that are applied directly to a deserialization block, SERDES, and converted into a 480-bit data bus.

Unwanted channels can be eliminated with a VME command [1], by controlling the MUX/MASK block. After a channel has been blocked out, the registers shall be cleared via VME so that no residual “1” bits are passed along and detected as hits.

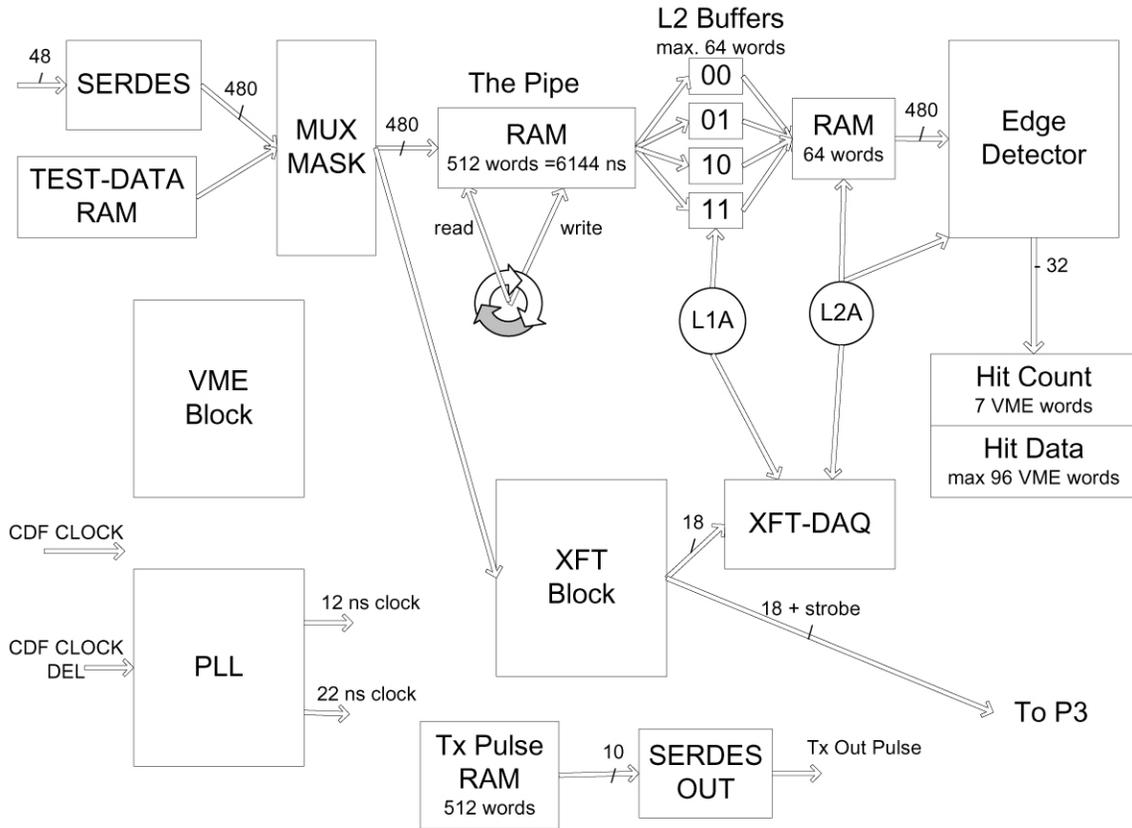


Figure 2. TDC Chip - Block Diagram

Upon a L1A pulse data passes through a 512x480 RAM, the Pipeline, and is written into one of the four L2 Buffers. The write and read addresses for the Pipeline come from the same, 9-bit, circular counter. The difference between those two values is controlled via VME and corresponds to the actual time difference between the L1A pulse and the moment at which the signal stored in the L2 Buffer reached the TDC chip. The writing of each L2 Buffer is independent; therefore the chip accepts L1A pulses on 132ns intervals. There is no data loss if one starts writing of a L2 Buffer before the writing of another one has ended. The number of words written in the L2 Buffer is set with a VME write command.

A L2A pulse transfers data from the selected L2 Buffer into another, identical memory and then to the Edge Detector [2] block. This second RAM is required in order to free the L2 Buffer for a new L1A pulse.

The Edge Detector receives a stream of 10-bit words for each channel and detects hits by analyzing every bit, starting with MSB of the first word. One "0" bit, followed by four "1" bits is considered a leading edge. One "1" bit followed by four "0" bits, is considered a trailing edge.

If the 0->1 transition comes before the beginning of the time range, in other words, if the 4 MSBs of the first word in the L2 Buffer are "1", the leading edge is 0x00.

If the 1->0 transition comes after the end of the time range, in other words, if a leading edge is found but no trailing edge, 0xFF is written as pulse width.

A maximum number of four hits can be detected for each channel.

The results are stored into two VME readout buffers, one for the number of hits and one for the actual hit data, as follows:

- Hit-Count: 4 bits are used to present the number of hits. Bit 3 represents the channel ON/OFF status and bits [2..0] the actual number of hits. The number of Hit-Count words to read after a L2A pulse is:
- $4\text{bits/channel} \times 48\text{channels/chip} + 1\text{header} = 7 \text{ VME words/chip}$;
- Hit-Data: Each hit is recorded as a pair of Leading Edge [7..0], Pulse Width [7..0]. The number of newly recorded hits varies after each L2A pulse; the maximum number of Hit-Data words to read after a L2A pulse is:
- $16\text{bits/hit} \times 4\text{hits/channel} \times 48\text{channels/chip} = 96 \text{ VME words/chip}$.

Since the Hit-Data VME readout buffer is never cleared during normal operation; it may include hits from old events at higher addresses than current hits.

See Run IIB TDC-II Address Space [1] for memory locations and word formats.

After each L2A pulse, the Edge Detector presents the number of new 32-bit Hit-Data words to the VME chip on board. The VME chip uses this info to control the Chain Block Transfer (CBLT). The number of Hit-Data words may also be read out via VME for testing purposes.

The same data stream that enters the Pipeline is applied to the XFT block that generates the trigger flags used by the eXtremely Fast Tracker (XFT) to identify tracks in the Central Outer Tracker (COT). The XFT block has two modes of operation, selectable with a VME command:

- New Style: six programmable time windows (time bins) [4] with the edges measured relative to a calibrated time interval, t_0 . The hit detection procedure is to find 4 sequential high cells in the input data stream (4.8ns) and to place this hit in one of the above mentioned time windows.
- Old Style: The first 3 time windows correspond to the PROMPT, NOTSURE and LATE windows [3] from the RunII-A design. The XFT output data: "XFT-Hit Bin 0" and "XFT-Hit Bin 1" [4] are the PROMPT and DELAYED bits [3] and follow the same truth table (Table2. page 5 from [3]).

The calibration of the t_0 time interval is achieved outside the chip using a VME controlled delay line for the CDF clock. In both modes of operation, the XFT flags are sent out beginning with the second, t_0 -delayed CDF clock.

The XFT block connects to a dedicated DAQ system similar to the main one. It has the same Pipeline/L2Buffers/VME-Readout Buffer structure and follows the same L1A/L2A sequence as the hit-data stream. The length of the XFT-L2 Buffers is also VME controlled. The XFT-DAQ is proposed to be used for testing purposes only. Also for testing, the XFT block is fitted with another simple VME Readout RAM that contains the current XFT flags. This RAM can be frozen and read out via VME.

The PLL block generates the 12 and 22 ns clocks used inside the chip. All the clocks are in sync with the t0-delayed CDF clock. The regular (un-delayed) CDF clock is also received and used only to latch the *B0, *BC, *L1A and *L2A pulses.

The SERDES OUT block generates one LVDS-Tx pulse pattern that is synchronous with and repeats after each CDF_B0 pulse. The number of successive pulses and the timing relative to CDF_B0 is controlled via VME by writing the content of Tx PULSE RAM.

The TDC chip also has a VME block for address decoding. See [1] for locations of all the memories and registers inside the chip as well as for the data formats.

4. TDC Chip – Specifications

- The TDC Chip is implemented on an Altera EP1S30F780C6 FPGA;
- 48 LVDS inputs with embedded differential termination;
- Sampling rate: 1.2 ns;
- Pipe-Line Size: 512 words (6,144 ns);
- Test-Data RAM Size: same as the Pipe-Line; test patterns are written via VME;
- L2 Buffer length: max 64 words (768 ns), VME controlled;
The four L2 Buffers are not VME accessible but each has a known power-up value for testing purposes.
- The chip accepts successive L1A pulses on successive CDF Clocks;
- On the next CDF Clock after L2A, the same L2 Buffer is available for writing;
- Edge detection: 1 hit = min four “1” bits (4.8ns) followed by min four “0”bits;
- Max number of hits/wire recorded: 4;
- Edge detection time: Hit-Count and Hit-Data are available for VME read-out ~7.25 us after L2A.
- XFT hit detection: 1 hit = min four “1” bits (4.8ns);
- Number of Time windows (bins): 6;
- The end of each bin is the beginning of the next and is programmable between 0 and 756 ns in steps of 12ns;
- XFT Bin time reference: programmable between 0 and 12ns in steps of 250ps by delaying the CDF Clock;

5. Block Diagram - TDC Board

The Block diagram of the TDC board is presented in Figure 3.

The complete schematic of the TDC Board is available on the web at: <http://edg.uchicago.edu/~bogdan/tdc/schematics.html>.

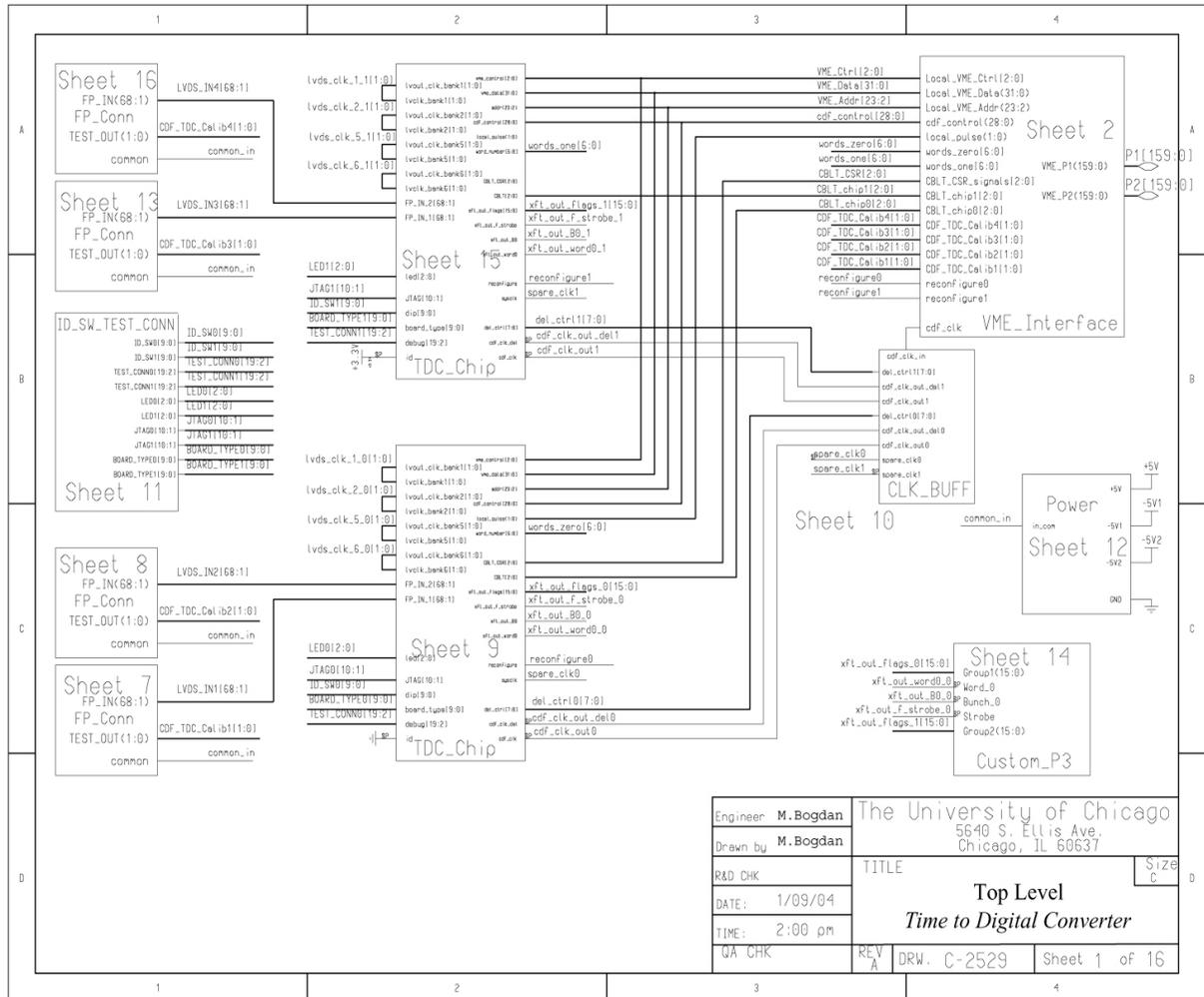


Figure 3. TDC Board - Block Diagram

The TDC Board receives 96 LVDS signals from the COT (48 for each TDC Chip). The signals are first applied to a repeater block, implemented with A1,...,4_1,...,6 and RN1,...,4_1,...,4 (Sheet 7,8,13 and 16). These LVDS repeaters have the same termination scheme as Rev.2A TDC. The midpoint voltage $V_{com} = 1.5V_{cc}$ is generated

with the linear regulator U561 from the +5Vcc power supply. The repeaters convert the COT signals into standard LVDS and pass them directly to the TDC chips.

The two TDC Chips: U_0 and U_1 (Sheet 9 and 15) receive the COT pulses and generate the Hit-Count and Hit-Data results in the VME Readout buffers.

The XFT flags are also generated and sent out via P3 (Sheet 14) through a set of ABT244 tri-state buffers: U2 through U6. The two FPGAs have identical design (“firmware”) and they both generate an output clock along with the XFT flags. However, only the clock from U_0 is sent via P3 and is used to strobe the XFT flags from both TDC chips.

Each TDC chip connects to two DIP-switches for board id and a triple LED pack on the front panel: LED0 and LED1.

The VME interface is implemented with an Altera EP20K100QC240-3V (Sheet 4). The VME and CDF signals are buffered with ABTE16245 (U17,...,U24). Since the TDC chips are not 5-V tolerant, an extra set of LVTTTL buffers (U27,...,U29) is used for the local VME Data bus and for CDF signals.

The VME chip connects to bits [15..0] of the VME data bus.

The CDF Clock from the backplane is first converted to TTL with a PECL receiver, U1, then phase-locked with ROBO, buffered with CLK_BUF and applied to U_0 and U_1 directly and also through a pair of programmable delay lines: DELAY_0 and DELAY_1.

The Board receives, via P2, the precision TDC PECL Calibration Pulses. They are passed through the 5V ECL Differential Multiplexer U564, on to the 5V ECL 1:2 Differential Buffer U14, converted to -5V ECL with U15, U16 (Sheet 2) and made available on each of the four front-panel connectors (Sheet 7,8,13,16). One has the option of switching between the Backplane Calibration Pulses to the locally generated calibration pulses (LVDS-Tx Output from the TDC Chip_0).

All the FPGAs on board are connected to a 20-pin header each for logic analyzer hook-up.

The power-up configuration files are stored in one Altera EPC2 device for the VME chip and in one Altera EPC16 device for each TDC chip. The FPGAs can also be configured via 10-pin JTAG connectors inside the board, same connectors used to change the default configuration files: JTAG_0, JTAG_1 and F2. For the two TDC Chips, the JTAG chains can be combined into one, front-panel accessible chain, by inserting shunts into: TCK_0,1; TDO_0_1; TMS_0,1; TDI_0,1.

6. Power

The TDC board uses +5 V/10A and -5V/1A from the backplane and generates the following voltages for internal use:

- +3.3 V/3A with the linear regulator: U562;
- +2.5 V/3A with the linear regulator: U7;
- +1.5V/20A with the DC/DC converter: DC_DC1.5V_0;
- +1.5V/3A with the linear regulator: U561;

- -3.3V/1.5A with the DC/DC converter: DC_DC_NEG3.3V. This power supply is provided with a front-panel resettable circuit breaker: BRK1.

The following fuses are required:

- F0_POS5V: 10A – main fuse;
- F1NEG5V1 or F2NEG5V2: 1A – negative backplane voltage fuse;
- F2_5V: 2A;
- F3_3V: 2A;
- FNEG3_3V: 2A;
- F1_5V_0: 5A;
- Fcommon: 1A.

The following LEDs are used as power indicators:

- LEDp-red: +5V (backplane);
- LEDp-yellow: +2.5V;
- LEDp-green: +3.3V;
- LEDneg-yellow: -5V (backplane);
- LEDneg-red: -3.3V.

7. CBLT Function

The TDC module permits Chained Block Transfer (CBLT) read commands [6].

There are two possible CBLT commands, as seen by the VME Crate CPU:

- Read block transfer from VME_Addr[31..27] = b"11111"(Slot 31). Hit_data words are read from every CBLT enabled board in the crate (0 to 192 words/board).
- Read block transfer from VME_Addr[31..27] = b"11110"(Slot 30). Hit_count words are read from every CBLT enabled board in the crate (14 words/board).

In this CBLT implementation, the TDC module closer to the Crate CPU is automatically considered "FIRST" in the chain with no setting required. Also, the CBLT is "ENABLED" by default. To set a module as "LAST", or to remove a module from the chain by disabling CBLT, one has to write a register in the module's VME Chip.

There are two ways to read out data from the TDC modules via CBLT:

1. Read out the hit-count words first, calculate how many data words need to be read from the hit data buffers and read out the exact number of words from those buffers.
2. Read out the hit count words and then initiate a CBLT from the hit data buffers for a number of words equal to the maximum number of words possible (not knowing how many data words need to be read from the hit data buffers).

At the end of the CBLT, the "LAST" TDC module will assert *BERR after the last word and the Crate CPU will terminate the read transaction.

In CBLT, *BERR is asserted only by the Last slave module in the chain to indicate the master that the last word has been transmitted. It is also used to reset the other slaves in the chain.

Note that *BERR is asserted by the "Last" TDC module at the end of every CBLT transaction, weather you read the exact number of words or not.

The same data is accessible for regular VME read-outs.

8. Reference

- [1] Mircea Bogdan and Harold Sanders, Run IIB TDC-II Address Space, 2004
- [2] Mary Heintz, How ED and ED48 Work – Rev.2, 6/28/04
<http://edg.uchicago.edu/~bogdan/tdc/misc/How ED and ED48 Work-Rev2.pdf>
- [3] Ken Bloom, et al, XTC: The COT TDC Mezzanine Card, June 11, 1998
- [4] BLW, Specifications for XFT Front end, March 4, 2003
- [5] T.Shaw and G. Sullivan, A Standard Front-End and Trigger VME bus Based Readout Crate for the CDF Upgrade – The CDF Readout Crate
CDF/DOC/TRIGGER/CDFR/2338, 1993.
- [6] ANSI/VITA 23-1998, Approved March 22, 1998.

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.