ASIC Labor
Heidelberg

Proposal for the LHCb Outer Tracker Front End Electronics

LHCb Outer Tracker technical note 2001-015

Harald Deppe, Franz Eisele, Martin Feuerstack-Raible,
Andre Srowig, Uwe Stange
Universität Heidelberg, Physikalisches Institut

Bart Hommels, Tom Sluijk
NIKHEF, Amsterdam

March 7, 2001

Abstract

A market survey on available TDCs for reading out the LHCb Outer Tracker has left over only one TDC, which is not optimal for this purpose. Hence, a new readout architecture which is based on a TDC to be developed anew has been defined. This system fits optimal the requirements of the LHCb Outer Tracker and also should be much cheaper. The system and its main issues are described in this paper.

1 Motivation for a new TDC development for the LHCb Outer Tracker

In the market survey for a TDC suitable for the digitisation and readout of the signals from the LHCb Outer Tracker it appeared that only the

“HPTDC” to be developed by CERN could meet the demands on the electronics specifications of LHCb[1].

The HPTDC is a general-purpose type TDC. However, the benefits of flexibility trade off against performance: the maximum allowed hit- and trigger rates are not guaranteed to be compatible with the LHCb specifications. By using less channels of the HPTDC the performance for each used channel increases; this feature is opening the opportunity to read out the LHCb Outer Tracker with the HPTDC.

Due to its data-driven architecture, the HPTDC is also not compliant to the general LHCb electronics front-end system. The state of the HPTDC is not predictable; it is dependent on the event size statistics. This means there is always a (small but not be excluded) risk of buffer overflows in the HPTDC.

Furthermore the HPTDC is not radiation hard so the chip cannot be placed directly on-detector. In practice this means that every preamplifier output channel needs to be transported off detector by means of a twisted-pair cable (differential signals).

The uncertainty in the predicted hitrates and the little performance headroom give enough reason to look into the feasibility of developing a new TDC for the LHCb Outer Tracker. A clock-driven TDC is to be preferred in order to be compatible with the overall LHCb front-end architecture. If a new TDC could be made in a radiation hard process then the new solution might also be more cost-effective because of the amount of cabling being greatly reduced. Less cabling also means less complicated mechanics and less access time for the stations in the magnet.

When comparing the data format of a data-driven architecture with an clock-driven architecture it is clear that the data driven variant needs to enclose redundant data with every hit. This results in a more compact data format for the clock-driven TDC. A data reduction factor of 3 to 4 is realistic.

In order to limit the amount of installed bandwidth, the trigger level 1 needs to be incorporated in the near-detector electronics. With a new clock-driven TDC the data format is more compact so it becomes feasible to transport the data off-detector at level 0. This has the advantage that the memories required for the level-1 buffering can be placed in the counting rooms, instead of the radiative environment of the LHCb cavity. Resulting from this the level 1 electronics do not have to have intrinsic protection against radiation-induced upsets and furthermore the level 1 boards are reachable at all times for maintenance, replacements or eventual upgrades.

2 Overview

For the Outer Tracker of the LHCb experiment new read-out electronics will be developed. The concept reflects the special needs of this experiment[1], namely

- High trigger rates up to 1MHz and relatively short maximum drift times
- High occupancy of up to 20%
- Large number of electronic channels but very little space for electronics and cabling, namely at the stations within the magnet, requiring very densely packed electronics
- Radiation hardness for a level of up to 1Mrad total dose at some stations

To solve these problems, a very compact electronics assembly is foreseen and a new TDC chip is developed. The new TDC differs from the ones available because its read-out architecture is a synchronous clock driven one instead of a data driven one. The new architecture is in line with the overall LHCb readout architecture and immune against changes in occupancy.

The chip is implemented in a 0.25μm CMOS technology which has been proven by the RD49 collaboration[2] to be radiation hard for up to 30Mrad total dose.

The very tight space constraint for the on-detector electronics is solved by integrating the TDC right onto the detector onto the same PCB that carries the discriminators.

By sending out data via high speed serial optical links[8] directly to the off-detector electronics the amount of electronics within the irradiated area of the detector is minimized, which should increase reliability and simplify maintenance. Also, no space consuming and costly flat ribbon cables are needed to guide data out of the detector to an intermediate station.

For the serializer chip we could use the GOL chip developed by Paulo Moreira et al.[6, 7]. It is implemented in the same radiation hard 0.25μm technology like the TDC chip and interfaces to it via a 32bit parallel bus.

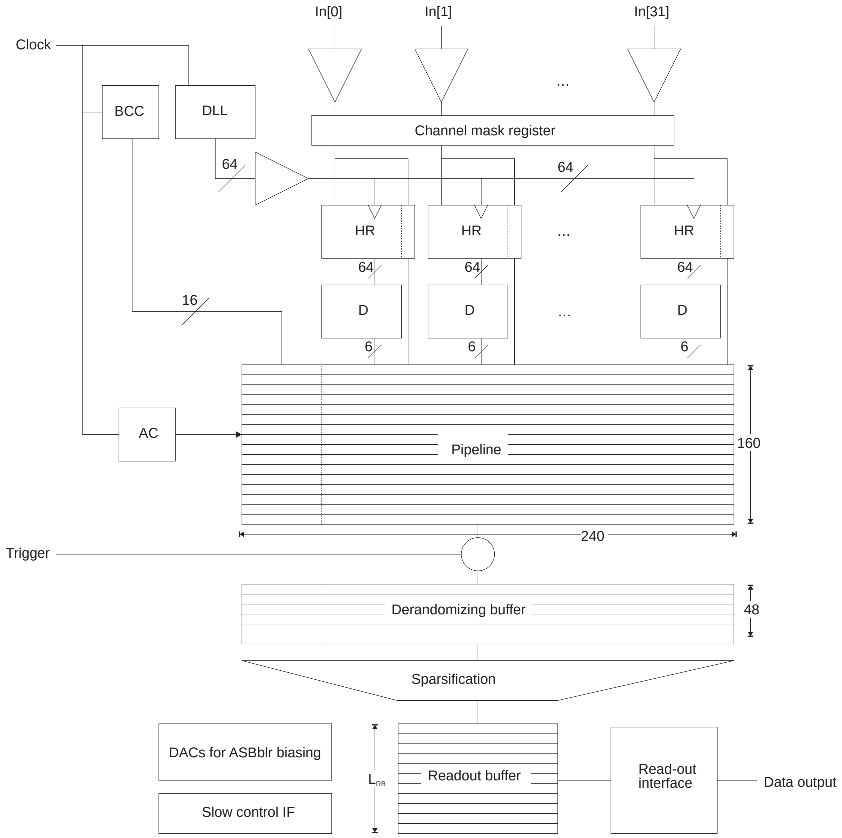

Figure 1: Schematic diagram of TDC for the LHCb Outer Tracker

3 The TDC chip

3.1 Architecture

The proposed TDC chip, which is to be developed anew, follows to a large extend the read-out scheme of LHCb, i. e. it has a synchronous clock driven scheme in contrast to the data driven schemes available in other TDCs. This architecture fits better into the LHCb environment, which has short drift times and high trigger rates and occupancies, than the data driven scheme, which is optimized for long drift times and small trigger rates and occupancies. The chip is implemented in a 0.25μm CMOS process with radiation hard layout technique. This approach has been proven to be radiation hard [2]. Figure 1 depicts the architecture of the TDC.

The TDC consists of a fine time measurement unit implemented as a calibrated delay locked loop (DLL)[4], a bunch crossing counter (BCC), a 32 channel pipeline with hit finder and derandomizing buffer, a zero suppression unit and read-out buffer and a read-out interface.

Since at every bunch crossing data are clocked into the pipeline for every channel, regardless of the channel having a hit or not, there is no intrinsic dependency on the occupancy within the chip. This dependancy is only introduced at the read-out buffer when data is sparsified before sent off the chip.

3.1.1 The DLL

The length of the delay locked loop is 32 stages. Since it is clocked with the LHC bunch crossing frequency of 40MHz it has a resolution of 25ns/32 = 0.78ns.

Since there is some probability that the rising edge of the discriminator output is coincident with a rising edge on the input of a hit register's flip-flop which than would become metastable, the decoder's output is unsure by one LSB. To maintain the specified resolution, each stage of the DLL, which consist of two inverters, has both inverters' outputs latched to the hit register. Hence the hit register is 6bit wide, but the final resolution is $25\text{ns}/2^5 \approx 0.78\text{ns}$ even when the drift time is a 6bit value.

The BCC is also clocked with the bunch crossing clock. Each time a hit occurs on any of the inputs, the current status of the DLL is latched into the hit register (HR) associated to the channel and a hit flag is set which indicates that the hit register contains valid data and which locks the hit register for the bunch crossing interval the hit has occurred. To prevent hot channels from increasing the amount of data to transmit, each channel can be masked.

3.1.2 The pipeline and the derandomizing buffer

At the end of this interval the encoded content of all hit registers and the content of the BCC is transferred into the pipeline location pointed to by the write pointer (WP), which is also forwarded with every bunch crossing.

Transferring data from the hit register to the pipeline also needs not to be protected against metastability, because actually the hit register is not clocked by the discriminator but the discriminator output is sampled by the hit register, i. e. the hit register is clocked by the DLL output. Hence, since the DLL is running synchronously to the bunch crossing clock, as the pipeline, the timing can be adjusted such that no metastability can occur.

The pipeline is actually a dual ported memory acting as a ring buffer of the length of the required latency, which is 160 according to the needs of LHCb[5]. Hence, after some time, data will be overwritten by new data. If a trigger arrives before the latency time of a data set has elapsed, these data are transferred into a derandomizing buffer.

3.1.3 The readout

To allow for a maximum drift time of more than the 25ns which are encoded by the DLL, either several events per trigger can be read out of the pipeline or data can be reorganized when being read. This last scheme is implemented to reduce the required readout bandwidth.

The principle of operation in the last case is that after a trigger a circuit called *hit scanner* scans for hits in a programmable number of consecutive data words (which cover the maximum drift time) in each channel and assembles a new data word called *extended drift time* which contains the drift time of the first hit found plus the bunch crossing offset from the time the trigger was received. Any further hit after the first, which is still within the scan window, is lost for this event. Since several bunch crossings can be scanned, the drift time word length is increased accordingly.

As mentioned above, if the application requires to collect all hits within the maximum drift time after the trigger, the hit scanner might not be used. Instead, a programmable number of consecutive events are read out of the chip upon a single trigger. Hence the number of events to be read out per trigger is also programmable. The drawback of this scheme is the much increased bandwidth necessary to read out the chip and also a bigger multi event buffer, since for each of the 16 possible triggers several events need to be stored. For this reason there is a restriction of up to two events read out per trigger which means the multi event buffer storage is 32 events deep, allowing for 16 triggers of 2 events each.

3.1.4 Hit scanner algorithm

Fig. 2 shows an example how the drift time scanner works. Two consecutive triggers are received by the chip (Trigger 0 and Trigger 1). Each trigger defines a sequence of 3 consecutive pipeline locations that are scanned for hits. The sequences may overlap as shown in the picture. For each channel the first hit and only the first hit per sequence is read from the pipeline. For example channel 0 has a single hit at bunch crossing 1. This is detected during the scan for trigger 0, but not in the scan for trigger 1, because the

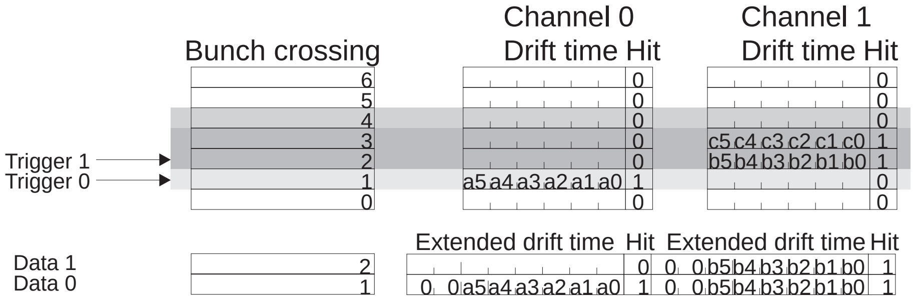

Figure 2: Example illustrating the hit scanner. Blank entries are "don't care".

scan window for this trigger starts later. Since for trigger 0 the hit was found in the first location of the window, the *extended drift time* is built by the original drift time plus an extension of value 0.

Channel 1 has two hits (at bunch crossing 2 and 3). The first hit is in the second location of the first window and thus is read out of the pipeline with an extension value of 1. For this window no further hits on this channel can be stored. The same hit is in the first location of the second window, hence it is read out again, but this time with a drift extension of 0. The second hit in channel 1 is not detected because for both triggers it is screened by the first hit. The first trigger that would detect this hit would be for bunch crossing 3.

The derandomizing buffer is a dual ported memory acting as a FIFO of length L_{RB}. Its contents are written into a sparsification stage that, if programmed so, removes the drift time information of channels which do not have the hit flag set and writes the result into a readout buffer.

The readout buffer is read by a read-out interface that either is a fast parallel bus interface of a fast serializing circuit which outputs a serial data stream of about 1Gbit/s. The options will be discussed in sect. 3.5.

3.2 Events lost by the TDC

Figure 3 shows results of a simulation of the deadtime behavior of the TDC (up to the end of the pipeline – no readout buffer is yet included) and the ASDblr discriminator in two different read-out modes at 10% occupancy. The y-axis shows the percentage of event caused by the various stages, i. e. the ASDblr deadtime, the data written into the pipeline synchronously to the bunch crossing and the hit search window width during readout. The

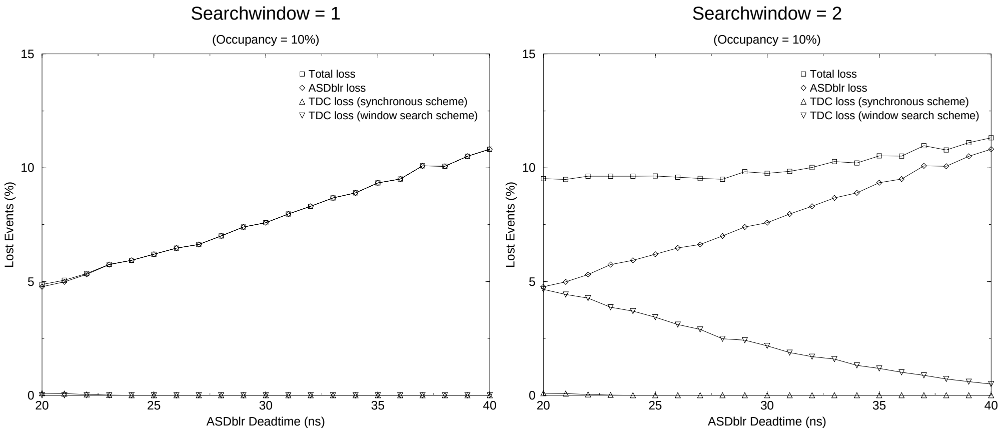

Figure 3: Deadtime introduced by the TDC with search window set to one (left) and to two bunch crossings (right)

x-axis is the deadtime of the ASDblr which is in the range of $20 \ldots 40$ns. The results scale with occupancy.

When the readout mode is set to a hit window search interval of one bunch crossing the total event loss is completely determined by the ASDblr deadtime. Even when this is only 20ns, only a sub-percentage contribution is originating from the TDC.

When the hit search window is set to two bunch crossings and when the ASDblr has the best assumable deadtime of 20ns, the events are equally lost by the TDC and the ASDblr. However, if the ASDblr deadtime increases to 40ns, again nearly all hits are lost by the discriminator chip and not by the TDC.

To conclude, we foresee the implementation of the extendable hit search window in the TDC and also the transmission of several events per trigger. This will give the system designer the opportunity to choose the optimum tradeoff between the system's deadtime and the required data bandwidth.

3.3 Estimation of the chip's size

The chips area is dominated by the hit registers and the memory.

The hit buffer must latch each of the inverters' output signals for any of the 32 channel. A simple D-Flip-Flop has a size of $16 \times 40\mu\mathrm{m}^2$. Hence the area occupied by the hitbuffer is $64 \times 32 \times (16\mu\mathrm{m} \times 40\mu\mathrm{m}) = 1.4\mathrm{mm}^2$

The memory consists of

- the pipeline of 32 channels $\times 160 \times 6 + 1$ bits

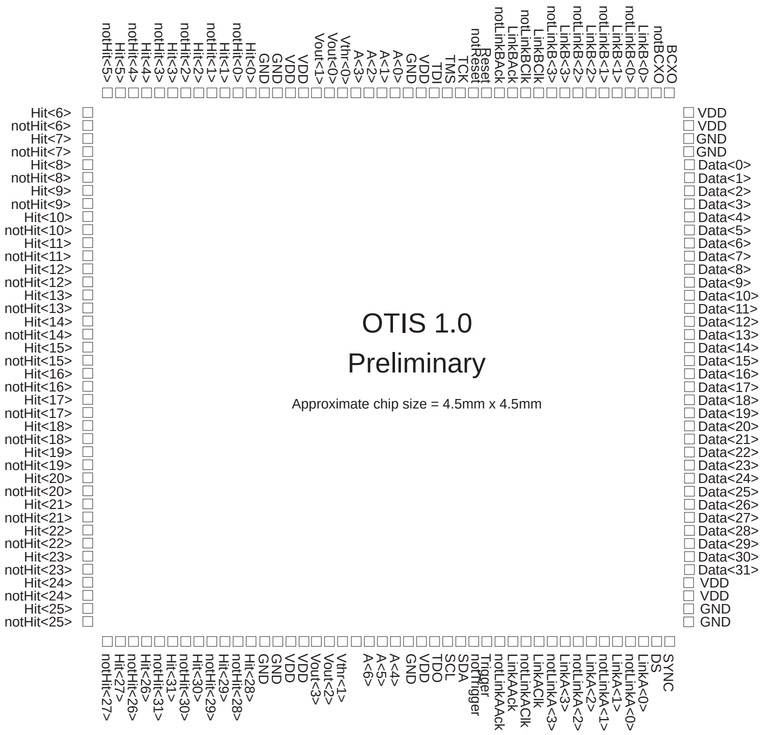

Figure 4: Preliminary pad layout of the TDC chip.

- the multi event buffer of 32 channels $\times 16 \times 3$ bits

which sums up to 49.920bits. The memory cell occupies an area of $10\mu \times 14\mu$, so the area of the pipeline plus the multi-event buffer is about 6.9mm^2

The read-out buffer may be of the same size at maximum. In total we get $1.4 + 2 \times 6.9 = 15.2$mm. Adding bias generators for the ASDblr, control etc. we can assume a chip size of about 20mm^2.

Fig 4 shows a preliminary pad layout of the chip. Due to the number of pads (about 130) the minimum size would be about 20mm^2, which fits nicely the above number.

3.4 Output reference data format

The reference data format of the TDC is shown in tables 1 (for single event per trigger read-out) and 2 (for multiple event per trigger read-out).

The first 16bit carry the chip ID, which is a 16bit number unique for the whole experiment programmed via the slow control interface of the chip.

Table 1: Reference output data format of the TDC for single bunch crossing read-out. n is the number of *extended drift time bits*, $N = m \times n$, where m is the number of hits in an event. Error checking codes or similar are not yet included because they are part of the media access protocol level.

Bit	$0 \ldots 15$	$16 \ldots 31$	$32 \ldots 63$	$64 \ldots 64 + n - 1$	$\ldots$	$\ldots 64 + N - 1$
Data	TDC-ID	BCC	Hit mask	Drift time[0]	$\ldots$	Drift time[m]

Table 2: Reference output data format of the TDC for multiple bunch crossing read-out. b is the number of bunch crossing read out, n is the number of *extended drift time bits*, $N = n \times \sum m_i$, where m_i is the number of hits in event i. Error checking codes or similar are not yet included because they are part of the media access protocol level.

Bit	$0 \ldots 15$	$16 \ldots 31$	$32 \ldots 63$	$\ldots$	$32 + b \times 32 - 1$
Data	TDC-ID	BCC	Hit mask[0]	$\ldots$	Hit mask[b]
Bit	$32 + b \times 32 \ldots$	$\ldots$			$\ldots 32 + b \times 32 + N - 1$
Data	Drift time[0]	$\ldots$			Drift time[N]

This number is followed by the 16bit bunch crossing number for the triggered event. A hit mask of 32 bit indicates whether a channel has had a hit detected by the hit scanner.

If the chip is programmed to send out sparsified data, depending on the number of bits set in the hit mask, a number of *extended drift time* values are following. If the chip is programmed to send unsparsified data, 32 *extended drift time* values are sent, though they have undefined values if they have no corresponding hit. For 100% occupancy both formats are the same.

If the chip is programmed to send multiple events per trigger, several hit masks (up to now up to 3, covering a maximum drift time of 75ns) are sent. All the drift time values are then appended in the same way as for the single event per trigger read-out.

Redundant information will be included into this format and also into the various on chip buffers to ensure data consistency even in a system with some bit error rate (e. g. caused by single event upset). This item will be addressed in future, when the media access protocol is defined.

Table 3: Reference data bandwidth needed per chip at 1MHz sustained trigger rate and various *extended drift time* widths, read-out modes and occupancies.

Occupancy	10%	20%	50%	100%
Ext. drift time 6bit	83.2Mbit/s	102.4Mbit/s	160Mbit/s	**256Mbit/s**
Ext. drift time 7bit	86.4Mbit/s	108.8Mbit/s	176Mbit/s	288Mbit/s
Ext. drift time 8bit	89.6Mbit/s	115.2Mbit/s	192Mbit/s	320Mbit/s
Ext. drift time 6bit, 2 events	134.4Mbit/s	172.8Mbit/s	**288Mbit/s**	480Mbit/s
Ext. drift time 6bit, 3 events	185.6Mbit/s	243.2Mbit/s	416Mbit/s	704Mbit/s

3.5 Sharing read-out lines to off-detector electronics for several chips

The mean occupancy of the LHCb Outer Tracker will be in the order of 10%, the maximum occupancy is about 20%. For the data format given in tables 1 and 2 and a sustained trigger rate of 1MHz this results in the reference data bandwidth needed per chip listed in table 3.

A single chip should always be able to send data at 100% occupancy with the minimum *extended drift time* width, i. e. 256Mbit/s. If 50% is counted as the worst case for a mean occupancy and if a drift time of less than 50ns is assumed, but one hit per bunch crossing is to be read out, this corresponds to an *extended drift time* width of 6bit and the read-out of 2 events per trigger. Hence, 288Mbit/s is a realistic maximum mean data rate.

The GOL chip is able to handle a data rate of 1.200Mbit/s pay load. Hence four TDCs could join a single fast serializer.

There are three alternatives for the topology of the TDC and fast serializer interconnects (fig. 5).

In a daisy chain chips are connected to their direct neighbours. Data are sent from left to right and each chip transmits data from its neighbours plus its own data. Since data streams are joined through the daisy chain, some kind of arbitration is needed. This is usually solved by introducing small buffers at the receivers (typically only a single word of transmission) and a flow control mechanism. The flow control is made with handshaking and can be realized in the same link technology like the data channels[1]. For this reason the daisy chain between the chips is bidirectional. Fair arbitration can be realized with a protocol running on the links. Concerning system

[1] This would also make the direction of the data stream configurable

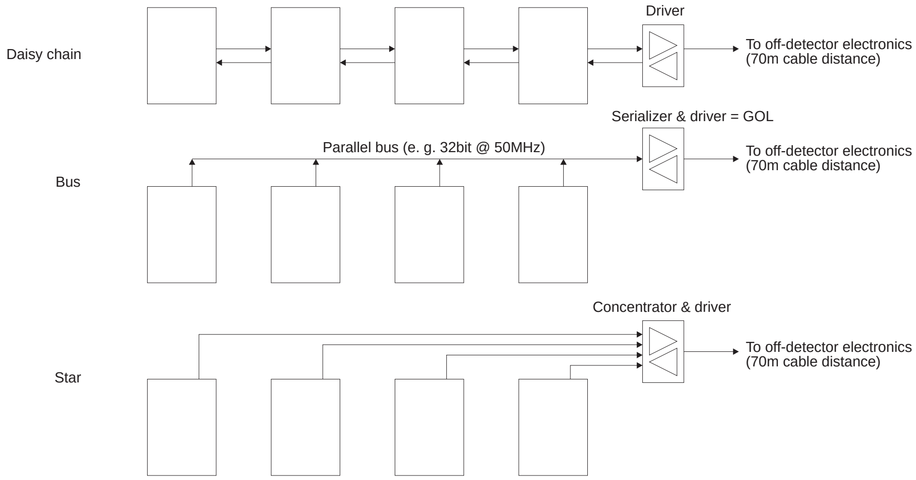
Daisy chain
Driver
To off-detector electronics (70m cable distance)
Serializer & driver = GOL
Parallel bus (e. g. 32bit @ 50MHz)
Bus
To off-detector electronics (70m cable distance)
Concentrator & driver
Star
To off-detector electronics (70m cable distance)

Figure 5: Alternative topologies to read out several TDCs via a single line

aspects the simple daisy chain scheme suffers from intolerance against single chip failure. If a chip is defect data of all chips upstream are lost also[2].

In a bus architecture several chips share a common bus line which transports data to the driver sending them to the off-detector electronics. Various arbitration schemes can be realized amongst which a token scheme which does "round robin" arbitration is the most simple one. The main problem with the bus architecture in our case is that the high speed serial links do not allow for several transmitters sharing a single line, mainly because of line termination problems at 1Gbit/s speed. This could be circumvented by implementing the bus in a different technology, e. g. a parallel bus which decreases the necessary clock frequency. In our case the 32bit interface of the GOL would run at about 40MHz. So this scheme introduces a significant amount of cabling or of data lines on the Signal distribution board discussed in sect. 4. Also the error tolerance is not good. A single defect chip may block the whole bus such that no other chip is able to send data.

The star topology is the one foreseen in all high speed networks like Gigabit ethernet etc.. This solves the line termination problems because there are only point to point to point connections as in the case of the daisy chain and it is relatively robust against single chip failures as long as the concentrator is operable. For our application it is not scalable very well, introduces a big cabling effort and still makes the development of a chip in

[2] A failsafe daisy chain scheme like implemented in the flow control e. g. of analogue pipeline chips can not be implemented here because of the large overhead introduced.

Table 4: Comparison of different read-out network topologies

	Daisy chain	Bus	Star
Rad hard sender	GOL	GOL	GOL
Rad hard receiver core needed	Implementation issue	No, bus must be parallel	Implementation issue
Arbitration	Flow control via daisy chain	Add. token network, round robin	Centrally
Cabling	Minimal	Large	Large
Reliability	Medium	Bad	Good

a radiation hard technology necessary, which multiplexes data to the GOL chip.

Table 4 summarizes the comparison of the different topologies.

To conclude, we choose the daisy chain solution, which offers reasonable reliability and minimum cabling effort. It is implemented in the way that the TDC chip has three bus interfaces:

- a 32bit parallel bus interface which is compatible with the GOL 32bit bus interface, i. e. the outputs are unipolar CMOS drivers with 2.5V signal swing. When the TDC is powered on, this interface is active and the last chip in the daisy chain connects vie this interface to the GOL chip.

- two 4bit differential LVDS link interfaces with which the TDCs within a daisy chain are connected. These interface are switched off after power off but can be started via the I2C bus slow control interface. The direction (transmitter or receiver) of each link port is individual programmable after startup. Hence the data flow direction of the daisy chain can be programmed.

Fig 6 shows the configuration for the Outer tracker.

3.6 Slow control interface of the chip

The slow control interface of the TDC is responsible to set up the chip before data taking starts and to observe the chip during operation.

An overview of the tasks is:

- Set up the chip's ID

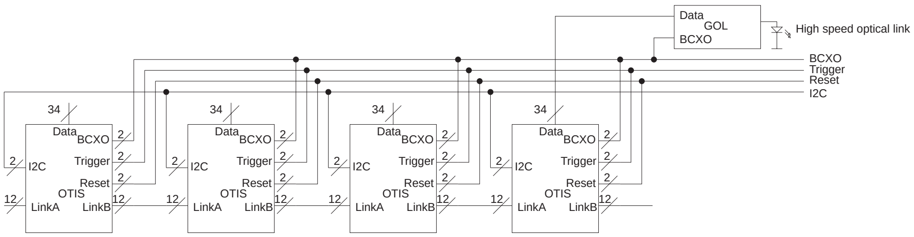

Figure 6: Daisy chained read-out for the LHCb Outer Tracker

- Choose *extended drift time* length and whether to sparsify data
- Program latency of the pipeline
- Program the DACs needed for the ASDblr biases.
- Detect on chip error conditions and store/signal them.

The common solution for these tasks is to use a standard bus. Implementations for the I2C bus[12] are available in the technology the TDC will be implemented in.

To solve the problem of spanning the long distance from the counting room to the On-Detector electronics we will follow the favorite solution for LHCb. In addition, we will evaluate the effort needed to implement a radiation hard CANbus node that would translate to I2C bus with the help of the ESA CANbus node implementation *Hurricane*[14].

4 On-detector electronics assembly

The read-out electronics consisting of the ASDblr chip, the TDC and the high speed serial link will reside directly on the detector. Data of several TDCs will be collected, which makes a connection between the TDCs necessary, and then send directly to the off detector electronics, which is about 70m cable length far away. There will only be a patch panel located just outside the concrete, no other electronics stations are needed.

The mechanics for the on-detector electronics assembly is depicted in fig. 7 on page 15. A tension board holds the straw tube wires and is fitted into a frame (e. g. aluminum). On one side of the double row of straws the tension board carries the HV decoupling capacitors and, if there is space left, also the HV spark protection circuitry. On the other side the HV bus plus the necessary components are mounted.

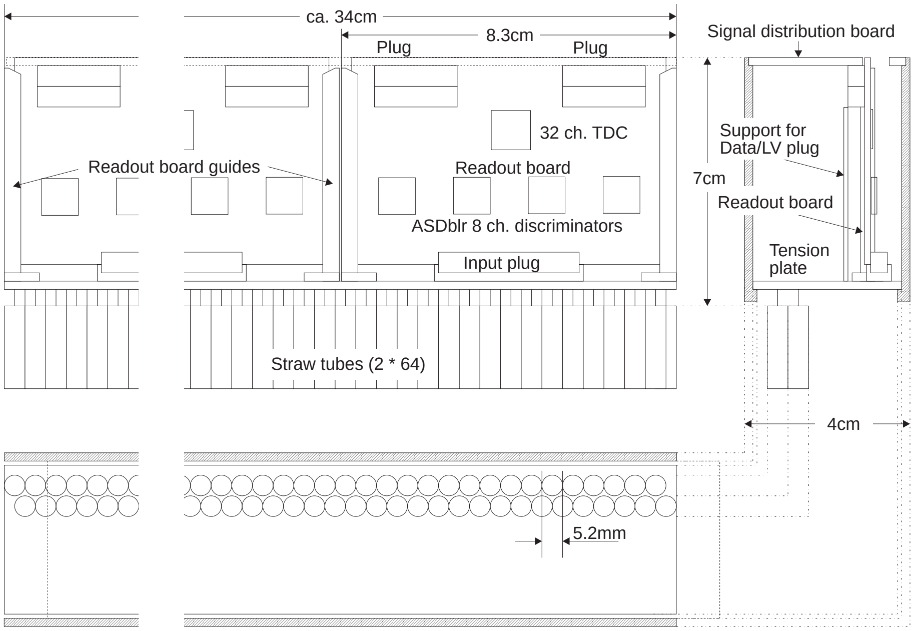

Figure 7: On-detector electronics assembly

4.1 The Frontend Board

The front end board integrates the ASDblr discriminators and the TDC chips. The available volume is about 4cm depth, 7cm height and the width is 64 × 5.2mm = 332mm, because of a straw tube pitch of 5.2mm and a maximum module size of 64 straws in 2 (staggered) rows.

Since due to the space constraints most likely chip-on-board (COB) technology will be used, the frontend board will only contain a single TDC and 4 ASDblr chips to achieve a good board yield. The board then serves 32 channels and has a size of about 8cm × 6cm.

4.2 The Signal Distribution Board

The signal distribution board guides the low voltage supply, the clock, trigger and reset lines, the slow control bus and the data lines to/from the readout boards. See table 5 for a list of signals distributed on this board. BCO, Trigger and Reset are LVDS signals, hence they need two wires each. For the daisy chain 4 differential data lines plus two differential control lines, in total 12 lines, are needed. For the parallel bus connection to the GOL

Table 5: List of signals on the signal distribution board for the parallel bus scenario and the high speed serial link scenario. "p. t. p." is a point to point connection between neighbours.

Name	Function	Topology	Number of lines Daisy chain
GND	Ground	bus	plane
VCC	2.5V supply	bus	plane
VDD	3.3V supply	bus	plane
BCO	Bunch crossing clock	bus	2
Trigger	Trigger	bus	2
Reset	System reset	bus	2
CAN	CAN bus	bus	2
LinkData	Data transport	p. t. p.	8
LinkCtrl	Flow control	p. t. p.	4
GOLData	Data bus	p. t. p.	32
GOLCtrl	GOL control	p. t. p.	2

chip 32 data lines plus 2 control lines are needed, all of them being unipolar.

For every group of four read-out boards a GOL chip and an optical transmitter is mounted on the signal distribution board. This is not shown in the picture.

5 Cabling to the off detector electronics

Though significant success has been made recently by the Gigabit Ethernet Initiative in sending 1Gbit/s via copper cable over a distance of 100m, up to now we intend to use optical fibres to send data from the detector to the control room.

Additionally there is the need to send the clock, trigger and control signals and the power supply to the signal distribution board.

Fig. 8 gives an overwiew of the Outer Tracker readout.

In the counting room there are a few Slow control boards, that drive the slow control lines to the Outer Tracker stations. These would be typically some kind of field bus, e. g. CANbus[13], which would be commercially available, or other developments for the same purpose, e. g. SPACbus from CERN. A reasonable number of slow control boards would be four per detector station, because the cabling for the top left, top right, bottom left and bottom right corner of the station woud be separate. Within a

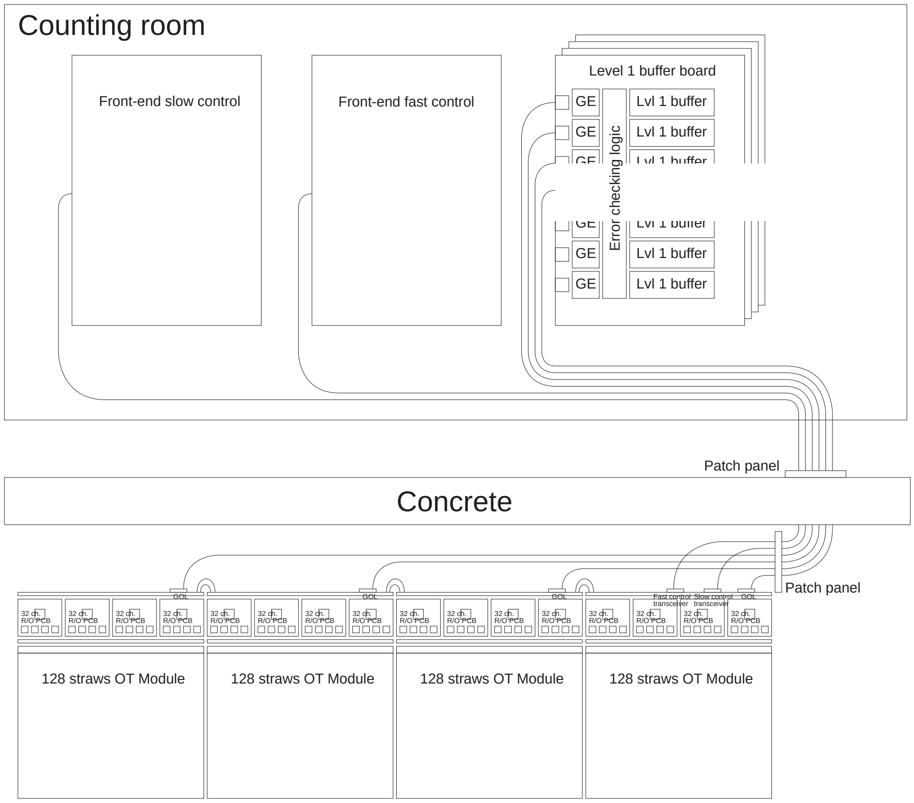

Figure 8: Overview of the Outer Tracker read-out

quarter station the signals end on a transceiver and then are transported from read-out PCB to read-out PCB via short cables.

The fast control is realized essentially the same way. The fast control boards reside also within the counting room. They house the TTCrx chips and distribute clock, trigger and reset signals to the on-detector electronics. Four such bards per detector station are reasonable here, also.

All cabling between the counting room end the on-detector station passes a patch panel outside the concrete. Spare cables between the patch pael and the counting room are introduced. There might be the need for another patch panel just outside of each detector station, due to installation issues.

Table 6 lists the required cabling. The choice of technology for BCO, Trigger and Reset, i. e. whether fibre or LVDS on copper, is still open.

There will be a patch panel located in vicinity of the concrete for all cables and fibres.

Table 6: List of cables between on- and off-detector

Name	Function	Cable type	Number of wires/fibres
GND	Ground	copper (thick)	1
VCC	2.5V supply	copper (thick)	1
VDD	3.3V supply	copper (thick)	1
BCO	Bunch crossing clock	copper/fibre	2/1
Trigger	Trigger	copper/fibre	2/1
Reset	System reset	copper/fibre	2/1
CAN	CAN bus	copper/fibre	2
Data	Data transport	fibre	1

6 Project schedule

The main tasks for the development of the Outer Tracker readout system are the chip development and the fast serial link, both the electrical version, which is partly covered by the chip development, and the optical version, which can be based on CMS developments.

6.1 TDC chip development

TDC core development proceeds as listed in table 7. The exact dates may vary because the submission dates for the MPW runs are not yet defined. We assume three to four dates per year and a turnaround time of two months.

6.2 Fast serial optical link development

Development of the fast serial optical link is based on the work for CMS and ATLAS. It can start with receiving Paulo Moreira's fast serializer in spring 2001. Components like VCSELs etc. that are already tested for CMS and ATLAS have to be acquired and an optical transmission line has to be set up in the laboratory. Tests have to be performed on the realizability of the electrical connections between the chips within a daisy chain.

6.3 PCB development

PCB development can start after receiving the first full size TDC prototype, i. e. about Autumn 2001. We foresee two prototypes for each critical board, i. e. the Readout PCB and the Signal distribution board.

Table 7: Chip design project schedule

Date	Task
Oct. 2000	Submission of 1st prototype of radiation hard TDC core with reduced resolution and without encoding
Oct. 2000	Start of development of fast serial receiver
Feb. 2001	Submission of 1st prototype of memory
Feb. 2001	Submission of 2nd prototype of radiation hard TDC core with full resolution and encoding
Mar. 2001	Start of tests with TDC core, 1st prototype
Apr. 2001	Start of tests with fast serializer of Paulo Moreira with commercial receivers.
Summer 2001	Start of tests with memory prototype
Summer 2001	Start of tests with TDC core, 2nd prototype
Late autumn 2001	1st prototype of complete TDC chip
Summer 2002	2nd prototype of complete TDC chip
Spring 2003	Mass production

References

[1] M. Feuerstack-Raible, B. Hommels, P. Kapusta, T. Sluijk, A. Zwart, "The Outer Tracker readout electronics", LHCb note 2001-013

[2] http://rd49.web.cern.ch/RD49/,
especially
http://www.cern.ch/RD49/RD49News/RD49StatusReport3.pdf

[3] ATLAS use of ASDblr:
http://www.quark.lu.se/ atlas/electronics/trt/chip_asdblr.html
Measurements for LHCb:
http://www.nikhef.nl/user/toms/lhcb/otfe/afe/deadtime/index.html

[4] Manuel Mota, Jorgen Christiansen, "A High-Resolution Time Interpolator Based on a Delay Locked Loop ad an RC Delay Line", IEEE Journal of solid state circuits, Vol. 34, No. 10, October 1999

[5] http://lhcb.cern.ch/electronics/html/key_parameters.htm

[6] A 1.25Gbit/s Serializer for LHC Data and Trigger Optical Links, Paulo Moreira et al., Proceedings of the 5th workshop on electronics for LHC, September 20-24, 2000

[7] http://proj-gol.web.cern.ch/proj-gol/

[8] For an overview of high speed serial links in HEP applications see http://hsi.web.cern.ch/HSI/

[9] For a tutorial, see for example http://www.iol.unh.edu/training/ge.html

[10] Agilent Technologies: "Low Cost Gigabit Rate Transmit/Receive Chip Set with TTL I/Os"

[11] http://hsi.web.cern.ch/HSI/dshs/

[12] Philips Semiconductors "The I2C-bus and how to use it (including specifications)", available via
http://www-us2.semiconductors.philips.com/i2c/facts/#specifications

[13] http://www.omegas.co.uk/CAN/

[14] ftp://ftp.estec.esa.nl/pub/ws/wsd/CAN/can.htm

[15] LHCb - A Large Hadron Collider Beauty Experiment for Precision Measurement of CP-Violation