

# 1 Detector Read Out and Data Handling

## 1.1 Trigger and Data Acquisition System

### 1.1.1 General Overview

The intense flux of a rare decay experiment, such as NA62, necessitates high-performance triggering and data acquisition. These systems must minimize dead time while maximizing data collection reliability. The NA48 trigger and data acquisition systems, designed more than 10 years ago, are unsuited to the task and can no longer be maintained for NA62. A unified trigger and data acquisition (TDAQ) system (1), which, as much as possible, assembles trigger information from readout-ready digitized data, addresses these requirements in a simple cost-effective manner.

The NA62 experiment consist of 12 sub-detector systems and several trigger and control systems, for a total channel count of less than 100 thousand. The GTK has the most channels (54,000), and the Liquid Krypton (LKr) calorimeter shares with it the highest raw data rate (19 GB/s). A summary of the number of channels and typical rates for the primary sub-detectors are shown in Table 1.

*Table 1: channel numbers and typical rates of primary sub-detectors.*

Sub-detector	Stations	Channels/station	Total channels	Hit rate (MHz)	Raw data rate (GB/s)
CEDAR	1	240	240	50	0.3
GTK	3	18'000	54'000	2'700	2.25
LAV (*)	12	320-512	4'992	11	0.3
CHANTI	1	276	276	2	0.04
STRAW	4	1'792	7'168	240	2.4
RICH	1	1'912	1'912	11	0.09
CHOD	1	128	128	12	0.1
IRC	1	20	20	4.2	0.04
LKr (**)	1	13'248	13'248	40	22
MUV	1	432	432	30	0.6
SAC	1	4	4	2.3	0.02

In the above table: “station” refers to a single unique physical location of electronics (this might include several boards or crates close together); the “Raw data rate” refers to the readout rate from the sub-detector boards to PCs (after a L0 trigger for all sub-detectors except LKr); such rates are quoted as pure payload rates, without any transport overhead being included. (\*) Each LAV PMT is read out by two electronics channels; the channel counts in this table are therefore twice the number of PMTs. (\*\*) LKr ADCs are continuously digitized at 40 MHz; the above rate correspond to assuming 8 samples around a trigger are read out at a L1 rate of 100 kHz without zero suppression.

A common coherent clock, with a frequency of approximately 40 MHz<sup>1</sup>, generated centrally by a single free-running high-stability oscillator, will be distributed optically to all systems through the Timing, Trigger and Control (TTC) system designed and used for LHC experiments (1). This “TTC clock” will be the common reference for time measurements<sup>2</sup> (section 1.1.4.1).

TTC optical links will also be used to send to each sub-system:

- a time-synchronous<sup>3</sup> L0 trigger accept pulse (section 1.1.4.2);
- a time-asynchronous L0 trigger type word (section 1.1.4.2);
- a time-synchronous start-of-burst and end-of-burst signals (section 1.1.4.3).

A common time scale is defined by a 32-bit **timestamp** word, with 25 ns LSB and covering the full duration of the interval between two consecutive SPS spills, plus an 8-bit **fine time** word, with 100 ps LSB. While the timestamp will be defined in each system by the phase-coherent distributed clock, each sub-system will locally generate by multiplication a properly locked reference for the fine time.

All clock counters should simultaneously reset at the start of each burst, using an appropriate synchronous command sent to all sub-systems through the TTC link before the actual arrival of the first beam particles. This will also define the origin of the time measurements for the burst. An end-of-burst signal should be sent in the same way some time after the end of the spill, defining the largest timestamp for the current spill. Its value should be recorded by each system and sent to the readout for logging, allowing (online and offline) a consistency check of the number of clock cycles counted by each system during each spill.

For test purposes, each sub-detector readout system should be able to run in a standalone mode, autonomously generating its own TTC signals (including L0 triggers) when not connected to the common system under global experiment control.

In principle, a “triggerless” system under software control, in which sub-detector data are unconditionally readout to PCs, would be the most flexible choice, but the rate and channel count of NA62 make this approach too expensive to implement. NA62 therefore has adopted a hardware lowest-level trigger labelled Level 0 (L0).

---

<sup>1</sup> Due to TTC system requirements linked to the LHC, the actual clock frequency is not exactly 40 MHz; in this document all references to e.g. “25ns” should be understood as the actual period of the main clock, (close, but not exactly equal, to 25ns).

<sup>2</sup> In a previous proposal for the LKr readout an additional very low-jitter 80 MHz clock, phase-coherent with the TTC clock, was considered to be used as sampling clock for flash ADCs: this could be distributed through the existing NA48 clock distribution system, which remains as an option.

<sup>3</sup> In this context “time-synchronous” denotes a signal occurring in a precisely defined 25ns time-slot with respect to its originating cause.

Following a L0 trigger, most sub-detectors will transfer data for one time-window to dedicated PCs, where a L1 trigger level will be implemented in software, and then to event-builder PCs where a software L2 trigger will be implemented.

The trigger hierarchy is thus made of three *logical* levels:

- a hardware **L0 trigger**, based on the input from a few sub-detectors; after a positive L0 is issued, data is readout from front-end electronics buffers to dedicated PCs (for most sub-detectors);
- a software **L1 trigger**, based on information computed **independently** by each **complete** sub-detector system, using data stored on dedicated PCs;
- a software **L2 trigger**, based on **assembled and (partially) reconstructed** events, in which complex correlations between information from different sub-detectors is possible, using data stored on the event building PC farm.

## 1.1.2 Trigger Logic and DAQ scheme

### 1.1.2.1 L0 Hardware Trigger

The hardware L0 trigger will be mainly based on input from the CHOD, the MUV, and the LKr, and optionally the RICH, the LAV, and the STRAW. The default (primary trigger) algorithm will be implemented to collect events with a single track in the CHOD, nothing in the MUV, and no more than one cluster in the LKr. The inclusion of other sub-detector information is possible, both to refine the primary trigger and to implement secondary triggers for control samples and different physics goals: for the main trigger, a multiplicity cut in the RICH and STRAW's may be able to augment the positive CHOD indicator, while the LAV might enhance photon and muon vetoing.

The CHOD will provide positive identification of a charged particle within the detector acceptance, reducing the rate due to  $K$  decays downstream of the final collimator. Hit multiplicity might also be used to select among multi-track events.

The third ("fast") plane of the MUV (MUV3) will veto muon events, *i.e.* the major background from  $K_{\mu 2}$  decays and the muon halo components from decays upstream of the final collimator. This rejection is the single largest rate reduction factor at the trigger level, and, consequently, its efficiency largely determines the L0 trigger rate (2). The geometrical acceptance of the MUV3 plane must exceed and include that of the positive track-identifying elements in the L0 trigger (e.g. the CHOD), and its online time resolution should be good enough (of order 1 ns), to avoid excessive random vetoing.

The LKr will also be used as veto in the L0 trigger, by allowing, in the primary trigger, no more than a single cluster (compatible with a charged pion EM shower). Simple quadrant energy deposition cuts could sufficiently reduce the  $K_{\pi 2}$  rate (2) in the primary trigger line, but online cluster counting with  $\approx 1$  ns time resolution may permit a better rejection and a more diverse set of physics triggers.

Hit multiplicity from the RICH might give a further contribution to the reduction of the background rates due to charged particles. Such reduction, however, may be rather limited, as the particle identification capabilities of the RICH cannot be easily exploited without momentum information, which is not available before correlation with the STRAW magnetic spectrometer at L2.

The inclusion of the LAV in the L0 trigger could also contribute to some further reduction of the  $K_{\pi 2}$  and muon halo backgrounds. Algorithms differentiating between a minimum ionizing particle and an electromagnetic interacting particle are under consideration and evaluation.

Additional information from other sub-detectors (including sub multiplicities in the STRAW) could increase the flexibility of the trigger, permitting alternative physics triggers. An algorithm requiring the number of MUV hits to be less than the CHOD multiplicity, the total number of clusters (in both LKr and LAV) to be less than the CHOD multiplicity, and no MIPs in the LAV would select  $K^+ \rightarrow \pi^+ \ell^+ \ell^-$  as well as  $K^+ \rightarrow \pi^+ \pi \nu \nu$  events.

Trigger primitives from sub-detectors involved in L0 will include both a timestamp and a fine time, in order to allow tight time matching. A L0 Trigger Processor (section 1.1.6) will time-match L0 trigger primitives issued by sub-detector trigger electronics and appropriately generate a trigger signal, which will be dispatched only with timestamps (25 ns time granularity), as it is expected that sub-detectors will readout data corresponding to (programmable) time windows longer than 25 ns.

Data from all sub-detectors will be stored in front-end buffers during L0 trigger evaluation. Upon reception of a positive L0 trigger, most sub-detectors will send their data to dedicated PCs within an adequate time window around a L0 timestamp. At every L0 trigger, each centrally-enabled sub-detector will respond to a L0 trigger by sending a data frame. The type and amount of data it contains may be different for different trigger types; a data frame may even be empty or indicate an error. No sub-detector may ignore a L0 trigger.

A maximum L0 trigger latency requirement implies that if no positive L0-accept signal is received within this period, the data can be discarded by the front-end buffers. A L0 trigger issued after the latency period is an error and should not happen under normal conditions; should it occur, each sub-detector will nevertheless reply with a properly formatted frame (most likely empty or indicating an error).

No data for untriggered events should be sent to PCs: downscaled events for control will be handled centrally by the L0 Trigger Processor, which will accept some events regardless of their failure to satisfy some specific trigger condition (another mechanism by which some sub-detectors might request that some specific event is forcibly collected is by issuing special trigger primitives).

An early and naive simulation of the detector hit rates and L0 trigger rates when using simple trigger cuts was produced (with an older detector set-up) using the fast FLYO MonteCarlo simulation, and is described in (1).

The maximum L0 trigger rate and latency are discussed in section 1.1.3.2. The possibility of increasing the L0 latency from 1ms to 6 or 7 ms should be foreseen by sub-systems, leaving open the possibility of implementing a smarter and more powerful (albeit possibly slower) L0 trigger scheme (see section 0).

### 1.1.2.2 L1/L2 Software Triggers

After a positive L0 trigger, all sub-detectors' data (with the exception of the LKr) associated with the L0 trigger timestamp, are moved to PCs for initial processing, which includes, at the very least, quality checks and reconstruction validating, as well as rudimentary pattern recognition.

A L1 trigger will require data quality verification, and then be based on simple correlations between *independently-computed conditions by single sub-detectors*, testing for the presence of a single charged pion in the detector (for the main trigger). A possible algorithm would be an equal odd number of tracks in the front and back of the STRAW spectrometer, no MIP or showering particle through the LAV, a hit multiplicity and pattern consistent with a charged pion in the RICH, and (optionally) a positive indication of at least one in-time kaon in the CEDAR.

Most detectors are expected to provide L1 Trigger primitives, if only in the form of rough data validation. One (single) PC (of perhaps several) associated with each sub-detector will be responsible for dispatching asynchronously the L1 trigger primitives for that sub-detector for *each* L0-triggered event to a central L1 Trigger Processor PC, based on complete sub-detector event data (which may have been merged from independent devices to include the entire sub-detector).

The L1 Trigger Processor will match these primitives and asynchronously issue a L1 decision, at which time the data will be transferred to the event-building farm (in the case of a positive L1) or discarded (in the case of a negative L1 verdict). The L1 decision will be delivered to the master PC of each sub-detector. This PC will dispatch the information to collect event data if they are split between different PCs. In case of positive L1 trigger decision, the information on the farm node to which event fragments have to be dispatched will be also delivered by the L1 Trigger Processor to the (master) sub-detector PC.

All L0-triggered events will get a L1 decision, and no data should be discarded until that decision has been received. The rate of the L1 trigger is not fixed, and there is no strict maximum latency for it, but L1 trigger evaluations are expected to terminate shortly after the end of each spill. No data for untriggered events should be sent to the event-building farm, as downscaled events for control will be handled by the L1 Trigger Processor, which will accept some events regardless of their failure to satisfy some specific trigger condition.

The possibility of running the experiment with more than one L1 Trigger Processor simultaneously active could also be foreseen: exactly one processor will handle one event.

With the data moved to the event-builder farm after a positive L1 trigger, crude full-event reconstruction can be done. Any event with an unaccompanied, single, identified charged pion will be accepted for the main trigger, but particle identification and 4-momentum consistency for other modes could also be checked.

A L2 trigger will be based on *correlations between different sub-detectors*. The information upon which these correlations are determined will be provided by event-building PC farms. Most sub-detector activity within an event time window will be at least partially reconstructed in the farm and made available for the L2 trigger decision.

For efficiency reasons the L2 trigger algorithms may be arranged into different hierarchical stages of conditional processing, *e.g.* by delaying time-consuming reconstruction until simpler conditions have been satisfied, or by running multiple reconstruction algorithms with different levels of refinement and complexity as needed. There is no dedicated L2 Trigger Processor, as each event will be dispatched for event-building and L2 triggering to different, dynamically chosen individual nodes in the farm.

All data associated with events satisfying the L2 trigger conditions will be logged to tape. In case L2 trigger conditions are not satisfied for an event, the data will be deleted (a fraction of failed events will be passed for purposes of monitoring and efficiency determination).

The rate of the L2 trigger is not fixed *a priori*, but will be determined by data logging capability. There is no maximum latency: L2 trigger computation can extend into the inter-spill period, but they should terminate before the next spill starts.

### 1.1.3 Requirements and Specifications

A detailed discussion of design requirements can be found in (1). Among more crucial ones ofr NA62 are reducing to a very low level any undetected partial failure of the readout for vetoing sub-detectors and avoiding any uncontrollable correlations between sub-detector trigger requirements.

Undetected vetoing failure due to data transmission errors can occur through two basic mechanisms:

- (a) failure to deliver data from some (part of a) sub-detector going unnoticed, or
- (b) time mis-alignment between data sent from one sub-detector and the others.

Once the data is within the processor farm, error checking mechanisms of the networking infrastructure can be exploited to limit the rate of occurrence of such errors, but particular care has to be taken in the first part of the data path, where custom electronics are used and data transfers occur between modules of different types.

Errors of type (a) are partly controlled by requiring that all sub-detectors (and all modules within a sub-detector readout system) always actively respond to a readout request (*i.e.* a L0 trigger), even if they have no data for that particular time region. Periodic DAQ integrity checks must also be performed asynchronously in an automatic way.

Errors of type (b) are controlled by continuous burst-level clock alignment checks, and event-by-event timestamp matching checks; the latter are important since data corruption in the timestamp is a single point of failure for vetoing (corruption in other parts of the data might not necessarily result in vetoing failure).

#### 1.1.3.1 Definitions

Unambiguous definitions of some TDAQ terms used throughout the rest of this document are presented here.

**Burst or spill:** the period of the SPS beam-delivery cycle, it is the basic data-taking time unit (event numbering and timestamping are relative to a burst, and restart from the beginning again at each new burst); the duration<sup>4</sup> is not specified and can usually vary in the range 1-20 s, but is (roughly) constant during each run; each sub-system should be capable of working with any value up to 50s..

**Run:** an arbitrary but convenient way of grouping a series of bursts taken under uniform data-collection conditions; runs cannot overlap in time.

---

<sup>4</sup> Includes both the spill time, when beam particles are hitting the target, and the subsequent inter-spill time, when they are not.

**Timestamp:** a 32-bit unsigned integer, relative to which all individual channel times are to be interpreted; defined by the L0 Trigger Processor, made available to each sub-system through a TTC receiver, and included in the event structure at all levels of data transport; uniquely related to the event number within a burst; the LSB equals the period of the master clock, roughly 25 ns ( $1/(40.07897 \text{ MHz}) \approx 24.951 \text{ ns}$ ), and the MSB is reserved, for a time range of about 53.6 s; some lower bits may be ignored when time matching for data extraction.

**Fine time:** the granularity for detector hit and L0 trigger primitive times; the least significant bit is 1/256 of the main clock period, roughly 100 ps (97.466 ps); not defined for an event, but event fine time will be available within the L0 Trigger Processor data frame.

**Event number:** the L0 trigger number (together with the burst ID) uniquely identifying a specific event; a 24-bit unsigned integer, corresponding to more than 16 s at the 1 MHz L0 trigger rate; provided by each TTC receiver; a difference in the correspondence between event number and timestamp among different sub-detectors indicates missed triggers (a severe error condition possibly forcing rejection of an entire burst's data). Event building from sub-detector data is based on the event number.

**Burst ID:** the UNIX time of a conveniently chosen instant in a SPS burst, uniquely identifying it and (together with the timestamp or event number) allows an event; a signed 32-bit number assigned by the farm management system (PC), broadcast to the entire TDAQ farm over the network, and entered into the data stream by sub-detector PCs.

**Run number:** a 32-bit unsigned integer identifying a continuous data-taking period under roughly homogeneous conditions; a higher-numbered run must contain higher-numbered burst IDs than a lower-numbered one.

**Trigger type:** a 32-bit unsigned integer indicating the L0 trigger type (the lowest 8 bits, assigned by the L0 Trigger Processor, see Table 9), L1 trigger type (the second 8 bits, assigned by the L1 Trigger Processor), and the L2 trigger type (the third 8, assigned by the Event Builder); the upper 8 bits are reserved (see Table 2).

*Table 2: Overall trigger type word format.*

Reserved	L2 trigger type	L1 trigger type	L0 trigger type
31	24 23	16 15	8 7 0

**Data block:** a series of aligned and possibly padded 32-bit words containing sub-detector specific information, in particular time information with respect to the event time, as defined by the timestamp.

### 1.1.3.2 Parameters

The values for the most important parameters of the TDAQ system are specified in the following table:

Table 3: Main TDAQ parameters.

Parameter	Value	Description
f(L0) max	1 MHz	Maximum average L0 trigger rate
$\Delta t(L0)$ min	75 ns	Minimum L0 trigger time separation
T(L0P) max	100 $\mu$ s	Maximum latency for generation of L0 Trigger primitives
T(L0) max	1 ms (*)	Maximum total L0 trigger latency
f(L1) max	100 kHz	Maximum average L1 trigger rate
T(L1) max	1 s	Maximum total latency for L1 trigger
F(L2) max	O(15 kHz)	Maximum average L2 trigger rate
T(L2) max	Spill period	Maximum total latency for L2 trigger

(\*) If possible, designing with a possible upgrade to higher latencies in mind would allow future implementation of different schemes for the L0 trigger generation.

### 1.1.3.3 Data Format

Each sub-detector and data source is uniquely identified by an 8-bit ID which is used to recognize its data, as indicated in Table 4.

Table 4: data source IDs.

Sub-detector	ID
CEDAR	0x04
GTK	0x08
CHANTI	0x0C
LAV	0x10
STRAW	0x14
CHOD	0x18
RICH	0x1C
IRC	0x20
LKr	0x24
MUV	0x28
SAC	0x2C
L0 Trigger Processor	0x40
L1 Trigger Processor	0x44
L2 event-building node	0x48
<i>Reserved</i>	0x4C
<i>Reserved</i>	0x80-0xFF

Besides detector data, most systems will also produce additional data as trigger primitives, which are identified by the lower bits of the ID, by adding 0x1 for L0 primitives and 0x2 for L1 primitives. For example, the LKr L0 trigger primitives will have ID 0x25, while the LAV L1 trigger primitives (all stations) will have ID 0x12.

The format of the data from each sub-detector is free until the data enter the event-builder farm after a L1 trigger. The format of the data frame (32-bit aligned) will be as follows:



Table 5: Data frame format after L1 trigger.

Word 0	Reserved		Total 32-bit word count (N+4)					
Word 1	Data source ID		Event number					
Word 2	Timestamp							
Word 3	Reserved							
Word 4 to N+3	Detector data block (N words)							
Bit	31	24	23	16	15	8	7	0

The total word count indicates the size (in 32-bit words) of the entire block (N+4), including headers and trailers; the maximum event data size is 60 MB per sub-system. The detector data block contains sub-detector specific data, 32-bit word aligned and possibly padded.

### 1.1.3.4 Readout Electronics

Front-end electronics (before digitization) are detector-specific and described in individual sub-detector chapters. Here we summarize general information about readout electronics (after digitization).

Sub-systems will receive a continuous 40-MHz clock, L0 trigger information (L0 accept trigger pulse and trigger type word), and start- and end-of-burst signals through the TTC, driven by the L0 Trigger Processor. These will be dispatched to individual sub-detectors by a TTC transmitter module (section 1.1.4), in principle allowing standalone running (for test or debugging). In practice, some smaller sub-detectors may be grouped to receive clock and L0 triggers from the same TTC module: in this case, they must function together (*e.g.* they will always be both included or excluded from a run).

Each sub-system is responsible for counting the number of clock pulses received between start- and end-of-burst signals, and for transmitting this count (as a data frame) upon request from the L0 Trigger Processor. Each sub-system must respond to every L0 trigger dispatch with a properly formatted data frame. Data can be transferred to respective sub-detector PCs only in response to appropriate trigger signals. Each sub-system must collect (at the end-of-burst, upon request from the L0 Trigger Processor) some status and monitoring information for monitoring and consistency checks, and send it in a special data frame. Finally, each sub-system must be able to drive a pair of lines to indicate that it is overloaded by data or in an error state (section 1.1.4.4): all such lines are merged at each sub-detector level and provided as inputs (one per sub-detector) to the L0 Trigger Processor.

### 1.1.3.5 L0 Trigger Sub-Detectors

Sub-detectors contributing to the L0 trigger will continuously evaluate their incoming data for the fulfilment of certain conditions (called “primitives” in TDAQ) and associated times. Times include both timestamp and fine time with typically a few nanoseconds intrinsic resolution, so that a number of lower bits of the fine time can be set as desired. Relative time offsets between sub-detectors should be corrected online so that the times of all trigger primitives are consistently aligned. Because each sub-detector has at most one connection to the L0 Trigger Processor (the Processor matches different sub-detector primitives, not primitives from the same sub-detector), data evaluation from multiple front-end cards will be centralized. The trigger primitive data will be packed into 8 bytes for every occurrence of some trigger primitive being satisfied, formatted as follows:

Table 6: L0 trigger primitive message format.

Word 0	Sub-detector ID	Reserved (0x0)	Primitive ID	Fine time
Word 1	Timestamp			
Bit	31	24 23	16 15	8 7 0

Inclusion of the sub-detector ID in the data packet offers the possibility of merging in a single link trigger primitives from multiple low-rate sub-detectors.

Each sub-detector can priority-encode primitives, so that when more than one is satisfied at a given time only a single piece of information is sent, but in this case evidence should remain within the primitive ID that such situation occurred<sup>5</sup> (this is required because the sub-detector trigger data for an event should indicate unambiguously ALL trigger primitives which were satisfied at that time). For primitives satisfied over an extended interval of time (of course shorter than the typical detector response time) a representative time should be determined.

Primitives will be sent asynchronously to the L0 Trigger Processor over Gigabit Ethernet (GbE) links; at the expected particle rate of 10 MHz, the maximum single-detector-to-L0 Trigger Processor bandwidth is 80 MB/s, which should be accommodated in a single GbE link. Primitive data will be packed in arbitrarily long lists, which need not be time-ordered, nor is there a fixed minimum time delay for transmission. Trigger primitives may therefore be packed to optimize bandwidth. A watchdog system might be required, however, to guarantee that under low-rate conditions primitives are not sent too late: primitives associated with a given time must reach the L0 Trigger Processor before the maximum L0 latency period  $T(LOP)_{max}$  elapses, after which the L0 Trigger Processor is free to make a final L0 trigger decision based on the primitives received so far. Reception of a primitive referring to a time older than  $T(LOP)_{max}$  is an error condition.

### 1.1.3.6 Higher-Level Triggers Sub-Detectors

All sub-detectors should have the capability to check and communicate the consistency of their data with decay mode- and sub-detector-specific trigger requirements, *i.e.* generating L1 primitives. All detectors involved in any L1 trigger decision must send L1 primitives to the L1 Trigger Processor in response to every L0 trigger, in the form of a 12-byte data packet shown below:

Table 7: L1 trigger primitive data format.

Word 0	Detector ID	Event number
Word 1	Timestamp	
Word 2	Primitive data	
Bit	31	24 23 16 15 8 7 0

<sup>5</sup> The simplest implementation of this is to associate one bit with a given primitive, but this would restrict the number of primitives to 8; since not all primitives will be independent, better encoding schemes can be used that allow the use of all 256 combinations, while retaining the information on condition overlaps.

Transmission of L1 primitive data can occur any time during the burst, without regard to event ordering. The L1 Trigger Processor will decide whether an event satisfies a L1 trigger only after corresponding L1 trigger primitives from ALL participating sub-detector L1 PCs have been received.

By the end-of-burst (plus some time to finish processing) the L1 Trigger Processor will have evaluated all L0-triggered events in the burst and dispatched L1 trigger messages to all sub-detector PCs, with the L0 and L1 fields of the trigger type word filled. A non-zero L1 trigger type field means the data for this event must be delivered to the event builder indicated in the packet (by its IP number). If the L1 trigger type field is empty, the last word of the packet is missing, and the data for this event should be purged. The format of the L1 Trigger Processor message is shown below:

*Table 8: L1 Trigger Processor message format.*

Word 0	Reserved		Event number					
Word 1	Timestamp							
Word 2	Trigger type word							
Word 3	Event builder ID (optional)							
Bit	31	24	23	16	15	8	7	0

No sub-detector may override L1 Trigger Processor decisions with regard to data handling. Keeping a (standard) fraction of events regardless of trigger decision will be a capability built into, and administered by, the L1 Trigger Processor.

## 1.1.4 Common Infrastructure

### 1.1.4.1 Clock Distribution

A single, free-running clock generator will deliver a continuous, high-precision, high-stability  $\approx 40$  MHz experiment clock to all sub-detectors<sup>6</sup> through the TTC system (1). This clock will drive all of NA62 timing systems and will run uninterrupted even when data-taking is not taking place. The TTC system (1) can encode on the same single-mode optical fibre via two time-domain multiplexed channels the clock, synchronous triggers pulses, and asynchronous commands. L0 trigger (from the L0 Trigger Processor) and start- and end-of-burst information (from SPS timing signals or faked by pulsers) will be distributed over the same link.

Twelve master clock partitions (each one dedicated to one or more sub-detectors) are foreseen, at present allocated as follows (the owning group is listed in parenthesis):

---

<sup>6</sup> If needed, the central clock generator might also drive the old NA48 clock system, to ensure that it maintains a stable relative phase relation with the experiment.

1. CEDAR (Birmingham)
2. GTK (Ferrara)
3. CHANTI (Napoli)
4. LAV (Roma/Frascati)
5. STRAW (CERN)
6. RICH (Perugia)
7. CHOD (Mainz/INR/IHEP)
8. IRC/SAC (Sofia)
9. LKr (CERN)
10. LKr/L0 (Roma Tor Vergata)
11. MUV (Mainz/IHEP/INR)
12. Spare

More than 12 sub-systems can be accommodated, but would share the clock/trigger distribution system with one of the above.

The master clock generator will distribute the 40 MHz clock signal to a fan-out card, which will drive in parallel 12 identical clock/trigger sub-systems housed in two master VME crates. Each clock/trigger sub-system, belonging to the corresponding sub-detector, comprises a modified version of the LTU module (3) designed for the ALICE experiment, and a TTCex (4) module with up to 10 identical optical outputs. Since each clock destination requires an individual fibre, passive optical splitters can be used to serve more destinations (up to 320 per TTCex module with a 1:32 splitter).

The clock from the main clock generator will be fanned out to a set of TTCex modules (4). Each sub-detector will receive up to 10 identical optical fibres from its TTCex module, and optical splitters can be used to reach more destinations (up to 320 per TTCex module).

Every electronics card requiring reference to the experiment time will be equipped with a TTC optical receiver, a TTCrx chip (5) which will extract the information from the optical signal and provide L0 trigger timestamps, and optionally a QPLL system (6) to reduce clock jitter. These might be integrated into each card, or, equivalently, the CERN-built TTCrq mezzanine board (7) can be used. Each sub-system interfaced to the TTC system should include a 32-bit timestamp counter, counting the number of clock cycles between start- and end-of-burst commands (also distributed by TTC).

#### 1.1.4.2 L0 Trigger Distribution

L0 triggers will be synchronous 25ns pulses<sup>7</sup>, and L0 trigger time information will be intrinsically encoded in the pulse occurrence time. The corresponding sub-detector timestamp will be in part provided by the TTCrx chip (5), as the start-of-burst signal will synchronously reset a local timestamp

---

<sup>7</sup> In TTC jargon, the time-synchronous trigger signal transmitted on “channel A” is called “L1 accept” (L1A); this signal will be used in NA62 to transmit the L0 trigger; the lowest trigger level is called Level 0 in NA62 as it will be the only one performed in hardware, to clearly set it apart from higher trigger levels, L1 and L2, which are performed in software.

counters. Differences in fibre length will cause fixed sub-detector offsets; the TTCrx chip can partially compensate for up to 80 m of fibre length difference (16 timestamp counts), but further adjustments may be necessary in individual sub-detector electronics.

The timestamp generated by the TTCrx for each L0 trigger is a 12-bit word, with 25 ns LSB, and therefore rolls over after 102.4  $\mu$ s. Coarse time information will be provided locally with a counter incremented by a suitably divided clock, derived from the master clock, and including an offset matching the timestamp offset; appending this coarse time to the TTCrx provided time will yield the complete timestamp.

Sub-detectors issue L0 trigger primitives asynchronously. The L0 Trigger Processor will re-synchronize them before driving the TTC transmitters, and L0 triggers are therefore dispatched in proper time order as synchronous pulses. Consecutive valid L0 triggers will be separated by a minimum of 3 timestamp counts (75 ns). The L0 Trigger Processor will synchronously dispatch trigger and burst information to multiple (NA62 version) LTU modules (3). Each such module will drive one TTCex optical transmitter module passing the information received from the L0TP, but it will also be able to internally generate triggers in case a sub-detector is running in standalone test mode.

Each central crate will require a VME processor for control and communication. Each processor will run as many DIM server daemons as the number of LTUs in the crate, so each LTU will be controlled by the daemon assigned to it. All daemons will be identical except for the identifier of the LTU crate they serve and their DIM identifier. All the servers in both crates will be approached independently. Each sub-detector group will have access to the processors, in order to be able to control its LTU module at any time, during the “development phase” of the experiment; during the “run phase”, the control of all LTUs will be centralized and handled by a common NA62 run control program. The VME processor configuration used for this “production system” should be considered the “NA62 standard configuration”, supported by the online group.

L0 trigger type information will also be encoded in 6 bits and transmitted through the TTC as an asynchronous command after each trigger pulse (“short B channel broadcast message” in TTC jargon): the two lower bits of the 8-bit word are reserved for start- and end-of-burst encoding (section 1.1.4.3). Five bits are available to encode 32 different types of physics and calibration triggers. The sixth bit is reserved for special commands for certain sub-systems to perform specific tasks and respond with appropriate data frames (remember that all sub-detectors must respond to all L0 triggers, if only with an empty data frame if they cannot handle the indicated trigger). The coding of the L0 trigger type word is shown below:

Table 9: L0 trigger type word encoding.

L0 trigger type	Trigger	Sub-detector action
0b0xxxxx	Physics trigger	Readout data (*)
0b100000	Synchronization	Send special frame
0b100001	<i>Reserved</i>	
0b100010	Start of burst	Enable data-taking, send special frame
0b100011	End of burst	Disable data-taking, readout end of burst data
0b100100	Choke on	Send special frame
0b100101	Choke off	Send special frame
0b100110	Error on	Send special frame
0b101111	Error off	Send special frame
0b101000	Monitoring	Readout monitoring data
0b101001	<i>Reserved</i>	
0b10101x	<i>Reserved</i>	
0b10110x	Random	Readout data (*)
0b10111x	<i>Reserved</i>	
0b11xxxx	Calibration	Readout data (*)

(\*) The type and amount of data to be read out can be different for different trigger types.

An example of such a special command is the **monitoring trigger**, which requests sub-detectors to send data frames containing monitoring information for inclusion in the data stream.

**Start-** and **end-of-burst triggers** define the valid data-taking time interval (and must occur after and the start- and end-of-burst hardware signals, respectively). These are the first and last triggers of each burst, and sub-detectors should respond to them by sending data frames as usual: the one for the end-of-burst trigger should contain monitoring data and statistics for the burst, which must include the timestamp count when the hardware end-of-burst signal was received, which will be checked for consistency offline, possibly taking into account relevant offsets).

Other special trigger codes might be defined, *e.g.* to distinguish between physics and calibration data-taking intervals.

Another special command is a **synchronization trigger**, in response to which all sub-detectors send a “sync frame”, formatted as a normal event. TDAQ will make use of such frames to monitor the live status of the entire chain through to the offline level, checking that all sub-systems were functional at least immediately before and after the trigger. The average frequency of these triggers will be chosen to allow adequate monitoring of TDAQ, but they will be issued aperiodically so as to avoid masking malfunctions linked to particular timestamp-related bit patterns.

The TTCrx chip will decode the trigger type word and distribute the information, initiating trigger-type-specific responses from front-end systems (*e.g.*, a sub-detector might reduce, or zero-suppress, data for some kinds of triggers but not for others, or issue an empty frame for calibration triggers related to a different sub-detector). A priority scheme will be programmed into the L0 Trigger Processor, so that

distinct, but simultaneously occurring, triggers (*e.g.* a physics trigger and a calibration trigger) can be handled properly.

Higher-level (L1/L2) triggers will define additional trigger types, requiring distinct processing algorithms, and they might also use the L0 trigger type word, *e.g.* for steering different processing algorithms.

### 1.1.4.3 SPS Interface

In NA62 the distribution of timing and clock will be done with the TTC system. WWE and a delayed EE will define the useful burst interval and will be distributed to all readout systems. The source of the clock and timing distribution, the central TTC crate, will be in the experimental area, and the SPS signals will be brought to this crate. Allowance must be made for user-defined timing sequences, and for part of the existing NA48 infrastructure which can be re-used after refurbishing the NIM logic<sup>8</sup>.

The TTC-defined “bunch counter reset” (BCRST) and “event counter reset” (ECRST) signals (the two lowest bits of the short broadcast message) will encode start-of-burst and end-of-burst commands as follows:

*Table 10: Start- and end-of-burst TTC encoding.*

<b>Command</b>	<b>BCRST</b>	<b>ECRST</b>
Start-of-burst	1	1
End-of-burst	1	0
<i>Reserved</i>	0	1

The start-of-burst command will be dispatched by the L0 Trigger Processor in response to a WWE signal from the SPS, before physics data-taking begins. The end-of-burst command will be dispatched by the L0 Trigger Processor in response to a delayed EE signal, after (possibly quite some time after) the beam extraction has finished. Start- and end-of-burst signals should always occur sequentially and

---

<sup>8</sup> NA62 re-uses partially the timing system of NA48, briefly reviewed here. The main inputs from the SPS timing distribution -- WWE (Warning of Warning of Ejection, about 1 s before the start-of-burst), WE (Warning of Ejection, just before the start-of-burst), and EE (End of Ejection, just after the end-of-burst) -- were regenerated by NIM logic as custom signals and fanned-out over copper wire by pseudo-differential drivers (CERN design) and audio-video pairs to four separate physical destinations in the ECN3 experimental area: the upstream region of the beam line (the tagger/KABES area), upstream of the detector close to the blue tube (the third drift chamber area), the electronics barrack, and the technical gallery. Each destination was equipped with a receiver module, a fan-out to which users connected, and a transmitter module that issued return signals for monitoring. Custom modules generated delayed copies (referred to as EC and ET) of the EE signal, and the down-counter of a four-fold scaler module generated a CLOCK RESET signal. All these signals were primarily employed to synchronize the readout and single-board VME processors, to create a burst gate for the hardware fast trigger logic, and to generate an in-burst interrupt to the slow control system to read the current in the drift chambers. The NIM logic could also fake a SPS sequence, useful for debugging the readout when the SPS was off.

be present, even in case of failure of the SPS timing signals, so that timestamp counters do not roll over their boundaries. An out-of-sequence command will cause the burst to be lost, and should be reported as a serious error. Each sub-system is responsible for counting the number of clock cycles received between a pair of start- and end-of-burst signals.

One arrangement which was found to be useful in NA48 was to extend the active data-taking time slightly more than the physical burst duration, and to use the additional time at the end (when beam is not present) to have calibration triggers; this required an additional timing signal to mark the end of the physical burst as opposed to the end of the data-taking burst. A similar scheme can be implemented in NA62 by providing some additional timing signal(s) to the L0 Trigger Processor only: sub-detectors need not and should not receive or depend on any other timing signals beyond the start- and end-of-burst ones discussed above.

The information on burst starts and ends might be required by at least a fraction of the experiment's PCs which perform different actions during the spill and inter-spill periods. The central system should take care of distributing such signals to these. Since this is not a time-critical task, such distribution is foreseen to be performed over the network, using the DIM software (8): one of the VME processors controlling the master clock crates will act as a DIM server for this purpose.

#### 1.1.4.4 Data Flow Choking and Errors

Each sub-system can drive two control lines, each a low-voltage differential signalling (LVDS) pair, transmitted on a single cable (per sub-detector<sup>9</sup>) with RJ45 connectors, informing the L0 Trigger Processor of possible trigger handling problems. Such signals indicate error conditions and *should not be used for data flow control*<sup>10</sup>.

The “choke” line will be driven high to indicate that a sub-detector is overloaded with data and approaching a situation in which it will not be able to accept further triggers without losing data. The L0 Trigger Processor will respond to choke signals and suspend L0 trigger dispatching as soon as possible. However, all sub-detectors, including the one driving the choke line, are still assumed to handle correctly all delivered triggers even when they have asserted the choke line, so that no data or transmitted L0 triggers are lost. An estimate of the response time of the L0 Trigger Processor to a choke or error conditions cannot be given until the implementation of such a device is known; however, since it is expected that the stopping of triggers will happen at the end of the synchronization board (see section 1.1.6), the response time can be assumed to be no longer than the physical time for transmitting the signals to the L0TP (1  $\mu$ s might be a safe figure).

---

<sup>9</sup> Merging signals from sub-cards is the responsibility of individual sub-detectors.

<sup>10</sup> This is the reason for avoiding the name “busy” or “XOFF” for the “choke” line.



If L0 trigger requests for some reason cannot be serviced, or if a L0 trigger timestamp exceeds the L0 trigger latency period and the data has been already purged, or for any other situation which could result in undetected loss of data, sub-detectors should drive high the “**error**” control line<sup>11</sup>.

The sub-detector driving the choke and/or error signal should keep it asserted at least until an acknowledgement is received from the L0 Trigger Processor in the form of a corresponding special trigger (“choke on” or “error on”), even if the situation leading to its assertion disappeared in the meantime.

The entire burst may be marked as unusable, but the L0 Trigger Processor will be capable of masking (i.e. ignoring) choke/error lines from individual sub-detectors.

In normal data-taking conditions neither the choke nor the error line should ever be asserted by any sub-detector. The size of the sub-detector buffers should be dimensioned in a way to sustain the maximum average rate, *including the rate fluctuations*.

### 1.1.5 Common TDC System

An early effort to find common (and possibly existing) solutions to common problems led to the identification of the TELL1 generic readout board (9), developed by EPFL Lausanne for LHCb, as the possible backbone for several applications, thus exploiting an existing product and reducing the amount of new hardware to be developed and maintained. The somewhat dated design was reviewed, and an improved version of the board was designed for NA62, with more powerful computing elements and much enlarged memory buffers.

The new readout and trigger board, called “TEL62” and electrically compatible with the original TELL1, is a 9U-format board which can house 4 independent mezzanine cards, each one served by a dedicated FPGA (“PP-FPGA”) with about 6Gb/s input bandwidth and 2GB of DDR2 dynamic RAM. A fifth FPGA (“SL-FPGA”) collects data from the four PP-FPGAs and drives another mezzanine card with four 1-Gigabit Ethernet (GbE) links. Other features include an on-board control PC (“CCPC”) for slow control (via a dedicated 100 Mbps Ethernet link), and a TTC receiver with a jitter-cleaning quartz-crystal-based phase-lock loop (QPLL), as well as user-definable connections for flow control. The card requires a special backplane with a single VME-like connector, which is only used for power. Apart from the general-purpose Gigabit Ethernet links, no inter-board communication mechanism is present, but a user-defined connector on the mother-board accepts a dedicated daughter-card which can be used for this purpose. Eighty to one hundred TEL62 boards will be produced for NA62.

LHCb developed, for the TELL1, a 16-channel, 40-MHz, 8-bit flash ADC mezzanine card (A-Rx), and a double-sized mezzanine card O-Rx (10) with 8 optical link receivers for CERN's Gigabit optical link transmitter (GOL). Many NA62 sub-detectors require no, or limited resolution on, pulse-height information, and so will use only TDC-based readout systems. Most of these will employ a NA62-built TDC mezzanine card (11).

---

<sup>11</sup> This includes the situation in which a sub-detector asserted the choke line but after a while, not seeing a pause in the trigger stream, is really no longer able to process additional events.

The NA62 mezzanine TDC board (“TDCB”) is equipped with 4 HPTDC chips (12) an FPGA (TDC Controller FPGA, or “TDCC-FPGA”), and 2 MB of static RAM. The board will service 128 input channels (LVDS) over 4 halogen-free SCSI-3 twisted-pair cables, measuring time and time-over-threshold with 100-ps LSB precision. Some capability to process data will be built-in. Up to 4 boards can be housed on the motherboard, for a total of 512 channels per mother-board. When equipped with TDC boards, two slots of crate space per mother-board are required.

Since time measurements are performed by the TDCs with respect to clock edges, several stages of filtering will be present to reduce the time jitter of the reference clock down to the  $\approx 50$  ps (RMS) level, without compromising the time resolution. It is possible to monitor one (fixed) channel per TDC chip (1/32) and/or drive it with the TDCC-FPGA for time calibration and debugging. The FPGA can also drive an additional LVDS pair, *e.g.* to trigger front-end board calibration pulses. Additionally, one (fixed) channel per chip (1/32) can receive a NIM signal from a front-panel LEMO connector, rather than from the input connector, for debugging and testing purposes.

The firmware for the TDC system will be split among the four TDCC-FPGAs (on the daughter-boards), the four PP-FPGAs, and the single SL-FPGA (on the mother-board). The code is loaded from on-board EPROMs at reset, and can be modified by accessing the board with a JTAG programming cable or with software via the on-board PC.

The **TDCC-FPGA** should:

- communicate with the CCPC by acting as an I2C slave, for receiving commands and configuration instructions;
- configure the four TDCs by acting as a JTAG master, according to data provided by the CCPC;
- control the TDCs at runtime, reading out data words when available and associating timestamps with them;
- monitor TDCs status, recording the occurrence of fatal errors (and their time of occurrence) in registers which can be read at the end of each burst;
- optionally pre-process TDC data words in local memory;
- provide data words on parallel and independent buses to the PP-FPGAs.

While HPTDCs have extensive multi-hit capability and internal trigger matching capabilities, their buffers are insufficient to store hits for the full latency of the L0 trigger. Data will therefore be read out continuously from the TDCs and buffered in the large RAM on the mother-board. Data time stamped within a programmable time window will be extracted after each L0 trigger.

When configured to run with 100-ps LSB, the TDC timestamp rolls over after  $51.2 \mu\text{s}$  (11 bits) and higher timestamp bits must be added. Although it would suffice to add the minimum number of bits to cover the maximum L0 latency, it is simpler to employ a full 31-bit timestamp word, thus requiring 20 additional (higher) bits. The time-matching feature of the HPTDC chips will be used by the controller FPGA to extract all hits belonging to a given time frame. By periodically triggering each TDC chip, and by setting the time window to the period between triggers, all hits are read out, automatically arranged in time-ordered frames. Each TDCC-FPGA will communicate independently with four TDC chips, triggering them all simultaneously, reading them out, adding upper timestamp bits, and storing all of this in associated FIFO buffers, from which the PP-FPGA will read using the same communication protocol. The TDCC-FPGA should monitor the filling of these FIFOs, and record instances of full

capacity, when possibly data were lost. Should there be no TDC data for a particular trigger (as will often be the case), no data frame will be generated.

For the TDC application, the **PP-FPGA** should:

- independently read TDC words from the four TDC boards and merge them into a single data stream;
- pre-process TDC words, in both a common and a sub-detector dependent way;
- split the TDC-generated time frames into convenient sub-frames (*e.g.* 25ns wide) and store TDC words in (off-chip) memory during the L0 trigger latency;
- (for sub-detector contributing to the L0 trigger) evaluate L0 trigger primitives and send them to the SL-FPGA;
- retrieve from memory a programmable number of data frames in response to a L0 trigger request from the SL-FPGA, and send them to the SL-FPGA.

For the TDC application, the **SL-FPGA** should:

- react to start- and end-of-burst commands, and dispatch time-critical commands to the PP-FPGAs;
- collect L0 triggers coming through the TTC;
- dispatch time stamped (and qualified) L0 trigger requests to PP-FPGAs;
- collect, merge, and format data packets from the four PP-FPGAs;
- prepare and send formatted multi-event packets (MEPs), together with dispatch control data, to the Ethernet link card for transmission to PCs;
- (for sub-detectors contributing to the L0 trigger) collect L0 primitives from the PP-FPGAs, merge them (perhaps with those from other boards), and send them to the L0 Trigger Processor or to another board.

The PP-FPGA will continuously move hits, timestamps, and errors independently from the four TDCC-FPGA FIFOs into other internal FIFOs, and then merge the hit frames into a common “input buffer”, making efficient use of available bandwidth. As the data are being stored, they can be monitored (*e.g.*, saving histograms of hit and error counts per TDC channel, to be read at the end of bursts), reformatted in smaller time frames, and processed (*e.g.*, pedestal can be subtracted, gains computed, etc.). For sub-detectors contributing to the L0 trigger, trigger primitives (*e.g.*, hit multiplicity in small fine-time bins) can be evaluated and temporarily stored in fine-time addressed buffers for transmission to the SL-FPGA.

To ease data retrieval, timestamp-addressed frames will be written into external DRAM memory, using a fixed page allocation. Because an integer number of frames around an L0 trigger timestamp will be read, the frame length should not be too long (25ns being the default). On the other hand, access to SDRAM is optimized with long writes; therefore frames might be grouped for writing, taking care to handle the roll-over when pages are re-used for new data. The number of words stored in each page will be kept in FPGA memory and cleared at roll-over time.

When the SL-FPGA transmits a L0 trigger request to the PP-FPGAs, it will also send timestamp and trigger-type information. The PP-FPGAs will extract the corresponding data frames from DRAM and pack them into FIFOs (together with a word count), from which the SL-FPGA will retrieve them. The

memory controller of the DRAM should arbitrate between write accesses (periodic frame storage) and read accesses (readout following a L0 trigger).

For sub-detectors contributing to the L0 trigger, each PP-FPGA should also transmit to the SL-FPGA trigger primitives information. A separate communication bus between the two types of FPGAs on a board will be used for this kind of data. The SL-FPGA will synchronously merge trigger primitives from the PP-FPGAs. Should a sub-detector require more than one board, these merged primitives must be further merged with similar information transmitted asynchronously from a previous board (e.g. using one of the GbE links), and again transmitted to the next board or to the L0 Trigger Processor. Such inter-board communication requires additional firmware support not required for sub-detectors using a single board (or those not involved in the L0 trigger decision).

The SL-FPGA will handle TTC communication. After receiving a start-of-burst signal from the TTC, it will distribute synchronous reset signals to each PP-FPGA, which in turn will synchronously reset TDC chips and timestamp counters in the TDCC-FPGA. When receiving a L0 trigger, it will retrieve the timestamp and trigger-type information and send L0 trigger requests and the corresponding trigger data to each PP-FPGA. When it receives an end-of-burst signal, it will record the last timestamp count reached.

For the sake of performance checks and the transmission of summary data at the end of bursts, the firmware should permit debugging (enabled only during tests) and monitoring (enabled during normal running, with no performance penalty).

A preliminary version of the firmware, nicknamed “TDCTEST” was developed in 2008-2009 to test the original TELL1 board, and was used in a 2009 RICH test beam. Lacking several required features, it was suitable for use only in simple standalone DAQ tests. It lacked: (a) large latency buffers for data storage; (b) timestamp generation, so that all data in a circular buffer were read out, irrespective of their time; (c) internal triggering, so that an externally generated trigger was required. The production version of the firmware, nicknamed “TDC”, will overcome the limitations of the TDCTEST version and will be suitable for a generic sub-detector's readout and L0 trigger primitives generation. Sub-detector groups will develop dedicated versions, “TDCRICH”, “TDCLAV”, etc., based on the TDC firmware. In most cases, only the PP-FPGA code will require modification.

The board will have other uses in NA62, as well, such as in the LKr L0 trigger system (section 1.1.15.5), and additional TELL1-compatible mezzanine cards will be developed, such as a pair of high-bandwidth asymmetric link cards and a digital data receiver card.

### **1.1.6 L0 Trigger Processor**

At the end of the latency period  $T(\text{LOP})_{\text{max}}$ , the L0 Trigger Processor (LOTP) will time-match the lists of trigger primitives it has received, checking whether appropriate (programmable) L0 trigger conditions have been satisfied within (flexible and programmable) overlapping time windows. It will simultaneously check multiple trigger conditions, with possibly different time constraints. It will generate down-scaled triggers for control purposes, as well as calibration and monitoring triggers.

Upon determining that conditions are satisfied, the L0 Trigger Processor will issue a L0 trigger. Candidate L0 triggers closer in time than a programmable range will be coalesced into a single trigger, since the time window within which sub-detectors will read data will be significantly larger than the L0 trigger time resolution. The list of remaining L0 trigger candidates, each having both a time tag and a

trigger type word, will be used to generate pulses with a fixed delay from the time of particle crossing, in order to generate a synchronous trigger signal (25ns resolution) to be dispatched by the TTC system (the so-called “L1A” signal in TTC jargon). For each L0 trigger sent in this way the corresponding 6-bit trigger word will follow asynchronously, to be recovered by the TTCx chip on each system.

In response to every L0 trigger, all sub-detectors will assign each L0-triggered event a sequential event number (uniquely linked to the L0 trigger timestamp, within a spill) and move data from within a certain (possibly trigger-dependent) time window around the trigger time in the form of event frames to permanent buffers (usually in sub-detector L1 PCs), where they can be erased only after a subsequent negative L1 trigger. These actions must be taken even when no data are available within the defined time-window, in which case the event frame will be empty.

Apart from inputs from the participating sub-detectors, the L0 Trigger Processor will also have other inputs, to be used as sources for “forced” synchronous triggers. In response to such pulses the L0TP will arrange, if possible, a special trigger to be sent after an appropriate (constant) time delay. This feature might be used in order to handle, *e.g.*, triggers related to calibration pulses in some sub-detector.

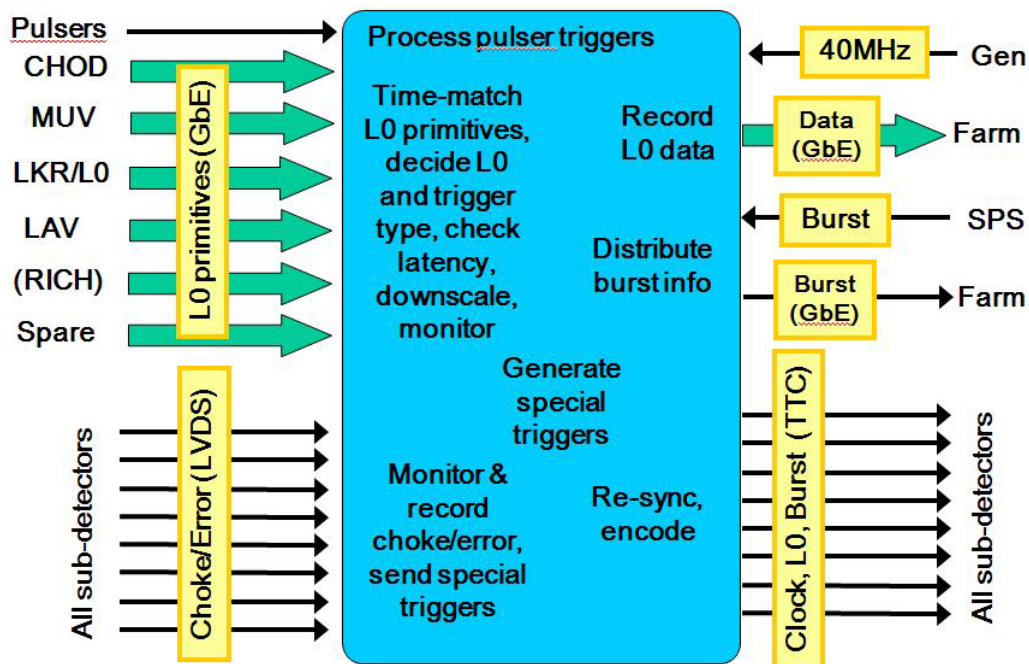


Figure 1: Logical scheme of L0 Trigger Processor.

The L0 Trigger Processor will continuously monitor maskable choke/error lines coming from each sub-system, logging the activity on each for inclusion in the data stream at the end of burst. In response to an unmasked choke signal, the L0 Trigger Processor will dispatch a special “choke on” trigger and cease issuing triggers. This special trigger is expected to be serviced as any other by the sub-detectors: the corresponding data frames will provide a means to check that no data were lost prior to the signal. When all (unmasked) choke lines go low, the L0 Trigger Processor will issue a special “choke off” trigger to resume normal operations.

In response to an unmasked error signal, the L0 Trigger Processor will dispatch a special “error on” trigger, in response to which sub-detectors may be required to send special monitoring data to

diagnose and debug (offline) the situation when the error condition occurred. When all (unmasked) error lines go low again, the L0 Trigger Processor will issue a special **“error off” trigger** before resuming normal operation.

The implementation of the L0 Trigger Processor is not yet defined. Several options are under consideration: a custom FPGA-based design is certainly possible, but the use of a high-performance PC with a real-time operating system, if proven to be feasible, might be simpler, more cost-effective, and easier to program and to maintain. In any case, a custom hardware part will be present, which will take care of re-synchronization of the L0 trigger pulses, and communication to the LTU boards; this will be implemented as a card with a PCI-Express interface, which easily adapts to any L0TP implementation.

### 1.1.7 CEDAR System

The CEDAR system will not participate in the formation of the L0 trigger, but might contribute to higher trigger levels.

The average kaon flux (neglecting accidentals) will be 50 MHz. Using 240 PMTs, the average singles rate will be about 5 MHz. Around 18 photons per kaon will be detected, while the probability of more than one photon traversing a given photocathode will be of order 1%.

The intrinsic time resolution of the PMTs is 300-400 ps, and a single-kaon time resolution of about 50 ps is required to suppress accidental background, so a minimum of 10 photons should be detected per kaon. Since dead time losses are of the order of 15 ns (11ns from NINO and 5 ns from the HPTDC), the double pulse resolution of the whole system is dominated by the electronics and will be no worse than 15 ns.

Both leading and trailing edge times, and thus the time-over-threshold, of the signal pulse will be measured. These times will be used to determine pulse amplitudes so as to correct for time-slewing induced by amplitude fluctuations and to discriminate against pile-up. Assuming a Gaussian shape for the PMT analog signal, one can show that edge times and time-over-threshold are linearly correlated. It is possible in principle to predict the correlation between time over threshold and amplitude, but for this purpose the contribution of electronic noise to the pulse shape must be negligible.

The readout for the CEDAR will be based on the common TDC/TEL62 system. The processing done in the TEL62 will include counting of the number of PMTs fired per spot and the number of fired spots, to later develop algorithms that use the multiplicity and the pattern of spots to suppress the background. It is also foreseen to record the number of photons in 1ns slots within the readout window, to allow for a precise time coincidence with the trigger signal. At any time, the TEL62 buffers should hold all the data corresponding to a time interval of 1 ms. Given an average of 9 hits per kaon per TEL62, with 32 bits per photon (leading and trailing edge measurements packed into a single word by the TDCs) and a kaon rate of 50 MHz, a maximum input of 40 Bytes is expected per kaon per TEL62, which corresponds to  $\approx 200$  KB in the buffer of 1ms, well below the size of the buffers available in a TEL62.

The NINO discriminator chip introduces a stretching time of 11ns, while the HPTDC has a dead time which can be set to 5ns. Nevertheless, the major limitation of this electronics is the finite size of the HPTDC hit channel buffer, which can sustain a maximum of 40 MHz over a set of 8 channels, or a maximum of 10 MHz for a single channel, whichever is lower (13). The 16 HPTDCs in a TDC/TEL62 board are arranged in sets of 8 channels which share some group channel buffers. By using only 2

channels for each set (1/4 of the total number available), the bandwidth will be reduced to a level guaranteeing a small probability of hit loss. Thus, only 32 channels per TDC board (128 channels per TEL62), and the CEDAR readout system will consist of 16 TDC boards and 2 TEL62 boards. The hit loss probability estimated in this configuration is 1%, which corresponds to a detection inefficiency per kaon of a few percent for a minimum of 10 photons distributed over at least 6 spots. The contribution of the readout system to the time resolution is estimated to be  $\approx 70$  ps, based on the performances of a prototype system used in RICH tests, a negligible contribution when compared with the intrinsic PMT resolution.

The leading and trailing edge measurements of each pulse can be combined in one word in the TDC, using 7 bits to measure the duration of the signal. Assuming an LSB corresponding to 200 ps, a maximum of 25ns can be measured as the signal length. The readout time window will be (-35 ns, +15 ns), for a total window of 50 ns. The window is not centered on the trigger value, to take into account the 25ns of the maximum signal length quoted above; 2-3 kaons are expected to appear in such time window. With an average of 18 hits per kaon, this corresponds to  $\approx 216$  Bytes in the readout window. Assuming a trigger rate of 1MHz, this corresponds to  $\approx 0.2$  GB/s of readout rate. Digital signals will be sent from the front-end electronics to the readout electronics through a few metres of high-quality twisted pair cable.

### 1.1.8 Gigatracker (GTK) System

As shown in the diagram in Figure 2, each GTK sensor is read out by 10 Giga Tracker ASICs (GTK-ASIC) whose output data flows continuously toward the GTK off-detector readout (GTK-RO) cards. These cards provide temporary data storage until the L0 trigger decision, upon which the GTK-RO cards extract trigger matched data from the on-board memory buffers and transmit the data to the on-line PC farm through Gigabit Ethernet switches.

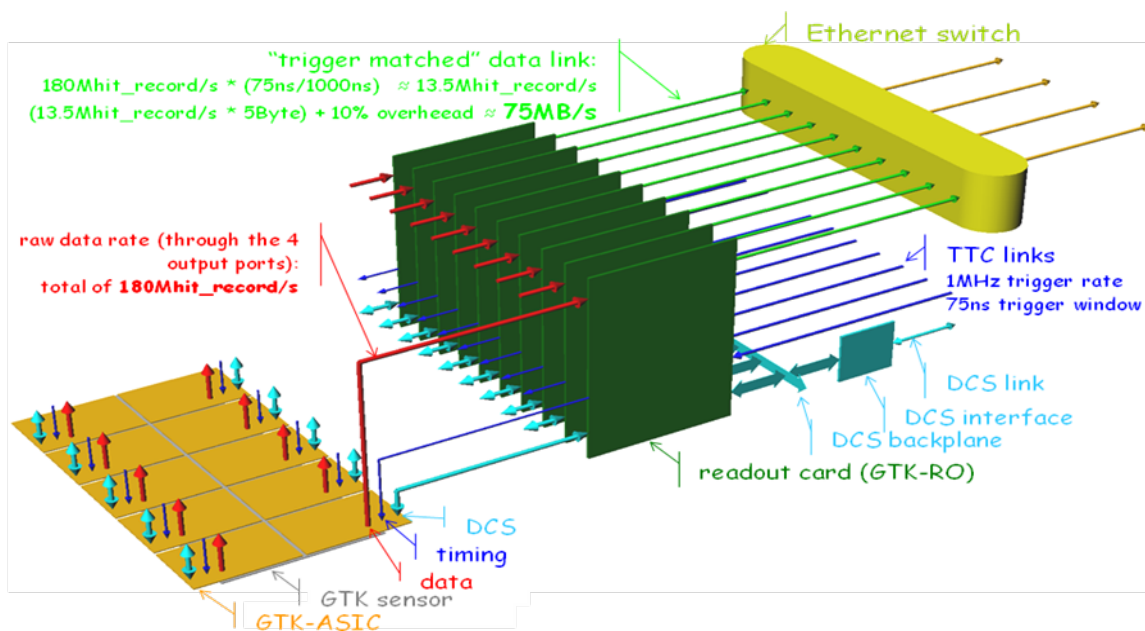


Figure 2 GTK Read-out block diagram.

The most relevant GTK sensor /ASIC parameters for the GTK off detector read-out are summarized below:

- maximum hit rate on centre pixel  $\approx 1.5 \text{ MHz/mm}^2$ , 140KHz/pixel;
- average hit rate per sensor plane  $\approx 750 \text{ MHz}$ ;
- data word width: 32 bit;
- average centre chip hit rate: 132 MHz;
- average centre chip data rate: 4.3 Gbit/s;
- design data rate (chip rate + contingency): 6 Gbit/s;
- serial links per chip: 2-4;
- readout window: 75ns.

The GTK read-out is based on the following scheme.

The GTK-ASICs continuously send periodic frames to the GTK-RO board. Each frame is made of one header, one hit record and one trailer. The header carries the GTK-ASIC ID and a rolling frame number, which is zeroed at the beginning of the data taking run. The hit record provides the local hit address within the pixel matrix and the time measurement which consists of a fine measurement derived from the TDC in the ASIC and a coarse measurement from the synchronous clock counter in the read-out ASIC. The range of the fine time is one clock period. The roll-over period of the coarse time determines the time interval covered by each output frame. If the EOC read-out architecture is chosen, a second fine time measurement for the trailing ToT signal is sent. The full dynamic TDC range extends to  $6.4 \mu\text{s}$ . The rolling frame number in the header is used to extend the system dynamic range of each hit to more than 10 ms. Each hit record is encoded in 4 bytes for the P-TDC architecture and 5 bytes for the EOC architecture to provide relative leading- and trailing-edge timing information. The trailer carries status information and a CRC-16 checksum.

	N. of bits	Resolution	Range
TAC	6 7	195 ps ( 56 ps r.m.s ) 98 ps ( 28 ps r.m.s. )	12.5 ns
Coarse counter	11 10	6.25 ns	12.8 $\mu\text{s}$ 6.4 $\mu\text{s}$
Frame counter	16	6.4 $\mu\text{s}$	838.9 ms 419.4 ms

*Figure 3 Hit format, resolution and range.*



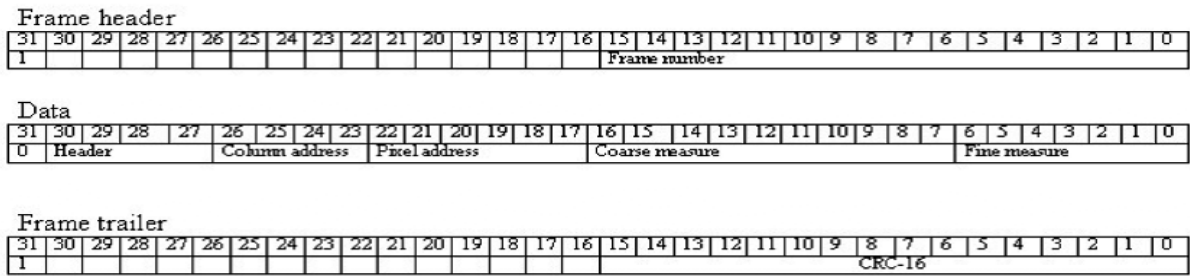


Figure 4 Header, data and trailer format.

Each GTK ASIC features 2 or 4 output ports, depending on final implementation. Depending on the ASIC internal distribution of the data flow and assuming an average hit rate of 4.5 MHz (3.3 MHz + 35 % contingency) per column, each output port transmits up to:  $45 * 10^6 * 5 * 8 * (10/8) = 2.25 \text{ Gb/s}$  including the 8b/10b encoding overhead but not including other sources of overhead like, for instance, the frame headers and trailers, of the order of 1%.

Figure 3 and Figure 4 show the hit word and frame format already implemented in the GTK P-TDC demonstrator [19], which easily can be adapted to the EOC architecture.

The requirements of the GTK read-out system follow from the above assumptions and are summarized in Figure 2.

To understand the data rates quoted in Figure 2 one must take into account the assumption that the L0 trigger matching is done on-line, *i.e.* not deferred to the “inter-spill” phase.

A 75MB/s data rate at the output of a GTK-RO card is achievable with a single GbE link if jumbo frames are used and if it is acceptable not to implement the full TCP/IP protocol. Two links could be otherwise foreseen, which would provide also some headroom in case the actual data rates would exceed the expected ones.

The following paragraph summarizes the read-out parameters for all three detector stations and for the configuration where 1 GTK-RO board serves one GTK-ASIC and 2 GbE links are available for one GTK-RO board.

- 750 MB/s of trigger matched data per GTK station
- 2.25 GB/s of trigger matched data for the entire Giga Tracker

Considering that 1 trigger-matched event contains, on average,  $180 \text{ Mhit/s} * 1 \mu\text{s} * 0.075 = 13.5$  hits per GTK-ASIC, and thus 135 hit records per GTK station, one finds

- 405 hits per event for the (3 stations) GTK and thus, assuming 5 Bytes per hit record and a 10% overhead:
- The average size of a GTK event  $\approx 2250$  Byte

One possible implementation of the GTK-RO board is outlined in the block diagram in Figure 5.

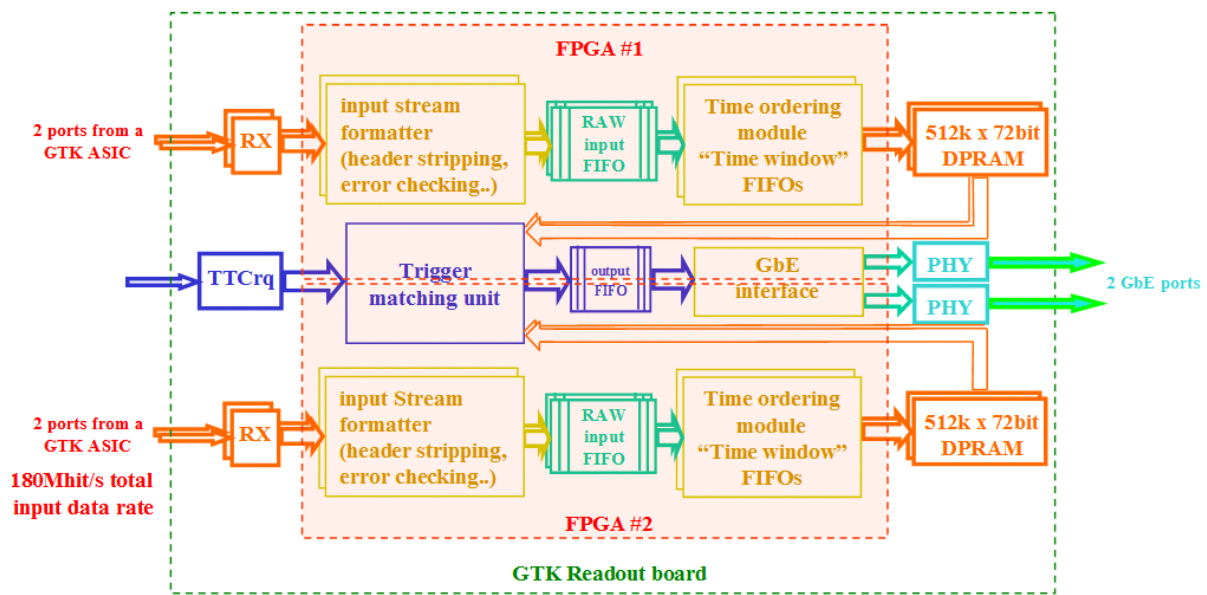


Figure 5 GTK read-out board block diagram

KEY PARTS:

RX: the choice of the deserializer depends on the standard chosen for serial TX on the final GTK ASIC.

TTCrq: TTCrx+QPLL mezzanine daughter card by CERN.

FPGA #1 - 2: Altera Stratix III, 780 pin, 480 I/O, 50k logic elements: EP3SL50F780C3N (≈ \$ 525 each).

DPRAM : Cypress synch DPRAM, 167Mhz, 512K x 72bit, 1.8V, CYD36S72V18.

Ethernet PHY #1 - 2: Marvell Alaska GbE PHY.

The key elements of any GTK-RO card architecture are the memory devices used for the temporary storage of the GTK data during the trigger latency time interval. A dual-port RAM (DPRAM) has completely independent read and write timing signals, R/W data buses and R/W address buses; the DPRAM considered in the diagram above has a capacity of 36 Mbits and operates at a maximum clock speed of 250 MHz. As shall be shown later, a GTK-RO card equipped with such devices may cope with a trigger latency of about 6.5 ms.

An alternative memory device such as the QDR-II+ Static RAM, features completely independent R/W timing signals, R/W data buses but a single address bus to be shared among the write and the read accesses. A QDR-II+ SRAM device can operate at frequencies up to 400 MHz, and thus a fast FPGA should be used to access the memory device by multiplexing the write and read concurrent accesses. QDR-II+ SRAM devices offer a higher density/price performance with respect to DPRAMs and become thus particularly effective large memory buffers are necessary to cope with an increased L0 trigger latency time. The following paragraph describes the operations of the individual blocks of the diagram sketched in Figure 5.

The **input stream formatter** controls the packets coming from the GTK-ASIC for CRC errors and checks the frame number against the one calculated on the GTK-RO card. It also strips off the header from the GTK-ASIC and transfers the hit data words into the Raw-Input FIFO appending an End of Frame (EOF)

marker to the data packet for 1 frame. Any error detected in the incoming packets would be coded into a specific field of the “EOF” marker to forward the error information.

The **time ordering module** (see Figure 6) is meant to facilitate the trigger-matching operation. The need for this module arises from the fact that the hit records coming from the GTK ASICs are not necessarily time-ordered. It seems then advisable to pre-order the input data in bins and store the content of each bin in a specific memory page of the buffer memory (be it DPRAM or QDR-II+). A bin is actually a small FIFO (Time-Window FIFO) memory inside the FPGA which is filled as the input data is extracted from the Raw-Input FIFO. Part of the coarse time measurement field of a particular hit record determines the FIFO it will be assigned to. If, for instance, a GTK data frame contains all the hits recorded by a GTK-ASIC in the previous 6.4  $\mu$ s, and if there are 16 Time-Window FIFOs, each of these would contain, when the Raw-Input FIFO is completely read out, all the hits recorded in a time window of 400 ns. One independent memory buffer is foreseen for each port. At the average input hit data rate for one GTK-ASIC port of  $\approx$  45 Mhits/s one 6.4  $\mu$ s frame will contain 288 hits. When one GTK data frame is fully processed, each of the 400 ns Time-Window FIFOs should contain on average 18 hits.

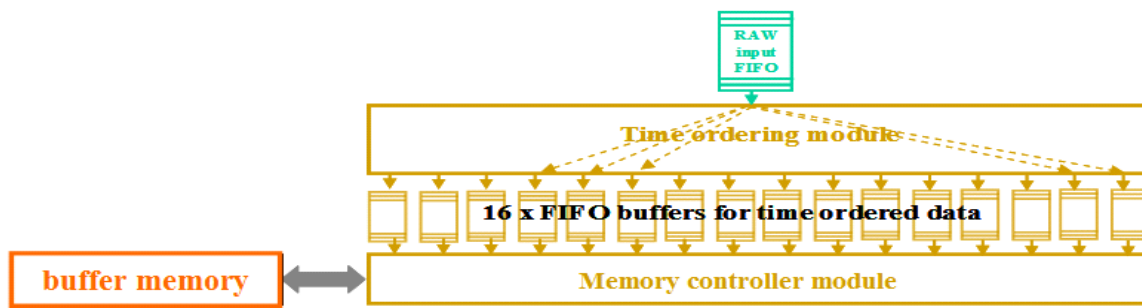


Figure 6 Block diagram of time ordering.

The contents of each Time-Window FIFO are then transferred to a specific memory page in the buffer memory. The frame number and the ID of the Time-Window FIFO (*i.e.* part of the coarse timing information in the hit record) determine the base address of the target memory page. When the content of a Time-Window FIFO is completely transferred to a memory page an End-Of-Page terminator is also stored. Taking into account that the DPRAM memories are 72 bits wide (36 bit is not enough for the EOC architecture) some of the coarse timing measurement of a hit is already encoded in the address, and 2 hit records could share the same memory buffer location.

Assuming that a memory page depth of 32 locations is reserved to store all the hits in a 400 ns time window (corresponding to a maximum capacity of 62 hit records , to be compared with the 18 expected in average) then a 512K \* 72 bit memory buffer would allow for a trigger latency of  $16384 * 0.4 \mu$ s  $\approx$  6.5 ms.

The interface to the memory buffer runs at a clock speed of 200 MHz, so up to 4 clock cycles are needed to fetch and assemble data in “double hit” words. One could then write 300 hit records (expected from each port in a 6.4  $\mu$ s frame interval) in approximately 3 $\mu$ s (20ns \* 150), which is comfortably shorter than the maximum 6.4  $\mu$ s frame period.

When processing an L0 trigger request, the trigger matching module must evaluate, from the known trigger latency, the memory page. Once the memory page is reached, 18 hits (the average content), can be scanned in  $\approx$  100ns (10 \* 10 ns), assuming that 2 clock cycles are needed to read and check for

the page terminator and assuming also that 2 hits are stored in each memory location. Thus the architecture outlined here seems capable of meeting the 1MHz L0 trigger rate.

The outcome of the trigger matching should be the extraction of an average of 13.5 hits from the whole GTK-ASIC, *i.e.* from the total of the 4 ports. Assuming that the hits are recorded in 5 Bytes, and taking into account an overhead of 10% for the event header, the average size of an event packet should be about 75 Bytes. Multi-event packets, similar to those built by the TEL62 board, should be assembled to optimize the bandwidth of the Gigabit Ethernet connection.

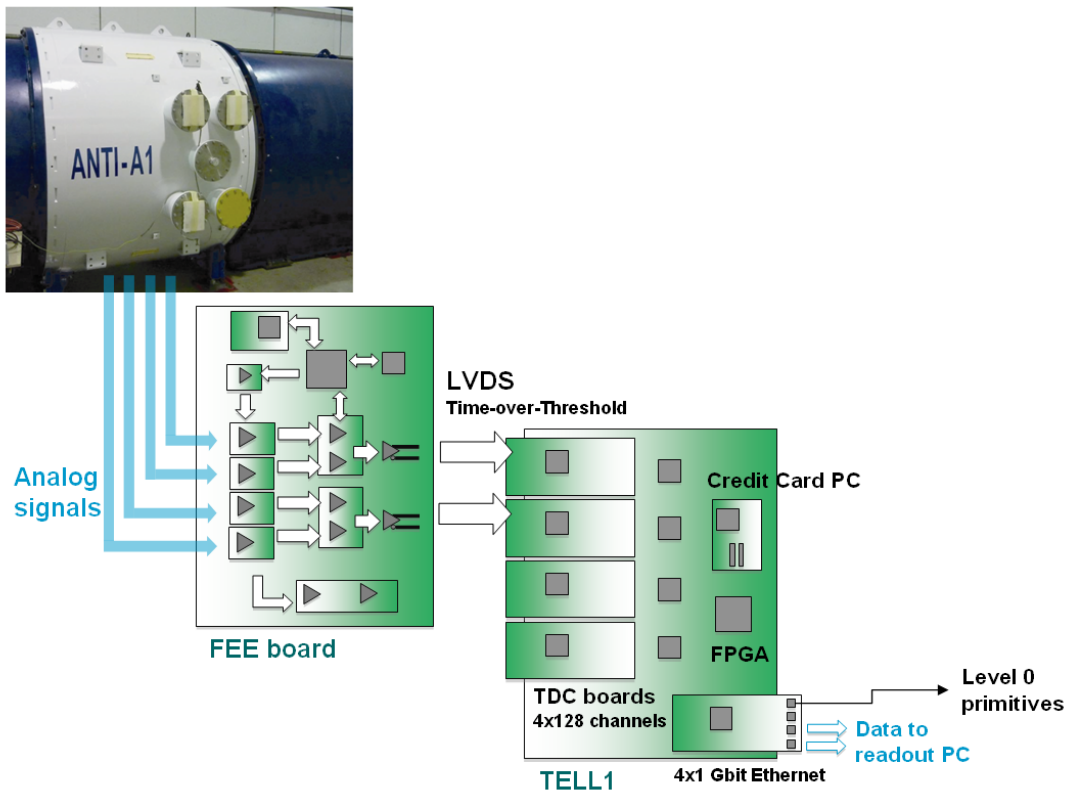


Figure 7: LAV readout scheme

### 1.1.9 LAV System

The output signals from the PMTs of each of the 12 stations are connected to Front-End Electronics (FEE) cards. These FEE cards discriminate the analogue signals from the PMTs generating digital signals of proper width (equal to the Time-Over-Threshold duration), using the differential LVDS standard.

Given the large dynamic range of the analogue signals, two digital lines are used for each PMT, with different thresholds, to allow a measurement of the signal rise time to correct for time slewing. The corrected time and the signal amplitude are reconstructed from the four (two leading, two trailing) measured times for each PMT pulse. L0 trigger primitives are also evaluated by the system, and sent to the Level 0 Trigger Processor. The general layout is shown in Figure 7.

Each FEE board has two 32-channel LVDS output connectors (a total of 128 wires), corresponding to two TDC board cables (half a TDC board). As mentioned, each PMT corresponds to two TDC signal pairs

corresponding to different thresholds, to allow time slewing correction and to ensure redundancy in case of TDC broken channels.

In Table 11 the number of PMTs per station, stations of each type, and the total number of TDC channels, FEE cards, TDC boards, and TEL62 boards for the whole LAV system are listed.

*Table 11: LAV channel counts.*

Station type	$N_{\text{PMT/station}}$	$N_{\text{stations}}$	$N_{\text{PMT}}$	$N_{\text{ch}}$	$N_{\text{FEE}}$	$N_{\text{TDCB}}$	$N_{\text{TELL1}}$
<b>Type 1</b>	160 (5x32)	5	800	1600	25	15	5
<b>Type 2</b>	240 (5x48)	3	720	1440	24	12	3
<b>Type 3</b>	240 (4x60)	3	720	1440	24	12	3
<b>Type 4</b>	256 (4x64)	1	256	512	8	4	1
<b>Total</b>		12	2496	4992	81	43	12

As the Table 11 shows, the LAV has about 2500 PMTs handled by about 5000 TDC channels. To equip the whole system about 90 FEE boards, 50 TDC boards, and 15 TEL62 are needed (including spares).

The dominant component of the rate in the LAV is due to muons, coming both from the beam halo and from K decays. The expected muon rates into the 12 LAV stations due to the beam halo are summarized in Table 12. Assuming that for each muon hit two 32-bit words are recorded from the TDC for each threshold, and that a muon is not firing more than seven blocks in each LAV station, the data rate is also indicated in the Table 12.

*Table 12: LAV data rates.*

	$N_{\text{PMT}}$	$N_{\text{fired}}$	$N_{\text{word/channel}}$	Rate	$N_{\text{bit}}$	Mbit/s
<b>LAV 1</b>	160	7	4+2	1.77 MHz	32	2378
<b>LAV total</b>	2496	7	4+2	11.2 MHz	32	15053
<b>LAV OR</b>	2496	< 25	4+2	4.13 MHz	32	19842

The number of firing blocks ( $N_{\text{fired}}$ ) has been over-estimated and we introduce a 50% extra hits, therefore the above figures for the data rate should be considered as upper limits. The rate per single station is lower than 1MHz on average, but is  $\approx 2$ MHz in the station LAV1. This rate can be translated into a single channel rate by assuming azimuthal symmetry for the muon halo, and the muon direction being parallel to the detector axis. In this approximation, for the LAV1, the halo rate is equally shared by 32 blocks of a single ring, resulting in a rate per single channel below 65 kHz.

The expected hit rate for a single channel is easily managed by both the FEE cards and the TDC boards. The total LAV data rate (< 20 Gbit/s) will be divided over the 12 TEL62 boards of the LAV system. Each TEL62 is equipped with four 1Gbit Ethernet interfaces and can therefore tolerate the expected rate.

The data will be sent to a commercial 48 in/48out 1Gbit Ethernet switch connected to the LAV sub-detector PCs.

The raw data coming from the TDC for each hit, namely the leading and trailing edge times for the two different thresholds, shall be converted into a slewing-corrected time ( $T_0$ ) and charge.

### Slewing correction

Define:  $T_L$  = leading edge time for the lower threshold,  $T_H$  = leading edge time for the higher threshold,  $T_0$  = event time extrapolated to zero amplitude (slewing-corrected),  $L_{THR}$  = lower threshold level,  $H_{THR}$  = higher threshold level. The correction is obtained with the following formula:

$$T_0 = T_L - L_{THR} \cdot \frac{T_H - T_L}{H_{THR} - L_{THR}}$$

### Charge computation

The computation of the charge is obtained according to a polynomial parameterization: after pairing the leading and trailing edge times, the 4<sup>th</sup> degree polynomial is computed to get the charge.

Both the above operations are performed on each hit and can therefore be executed in parallel in the PP FPGAs (described in Common TDC System)

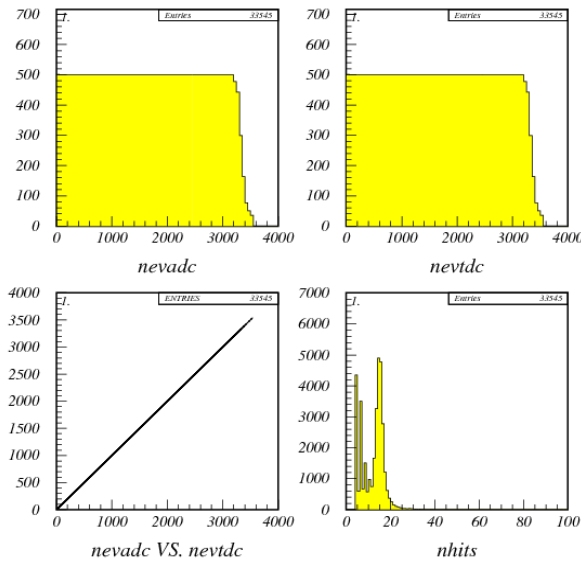


Figure 8: LAV hit multiplicities.

### L0 Trigger

The raw information available in the LAV system at L0 trigger evaluation time are the times and charges of single hits in each block of a single station. These are all collected in the TEL62 board serving one station and can be elaborated by the on-board FPGAs to produce for the L0 processor simple primitives containing ring and station summary informations, including the type of particle.

As each station is composed by 4 or 5 ring-shaped layers of crystals, we can combine data to produce simple summary variables containing ring and station information:

### Ring primitives (circle of blocks)

- $E_{bl}$  = reconstructed charge in a single block
- $E_{ring}$  = charge sum of all the blocks in the ring
- $N_{ring}$  = number of blocks above threshold in the ring.

### Station primitives

- $E_{tot}$  = total charge of all the blocks in the station
- $N_{tot}$  = total number of blocks above threshold in the station
- $N_{cl}$  = total number of clusters, using a proximity algorithm.

Using the above informations and very simple cuts, two LAV triggers primitives can be constructed:

**MIP trigger:** to identify MIPs ( $\mu$  or  $\pi$ ) and to distinguish them from photons and electrons:

- $N_{ring} \leq 2$  for each of the five or four rings
- $3 \leq N_{tot} \leq 7$
- $E_{bl} < 250$  MeV for each block over threshold
- $E_{ring(i)} / E_{ring(i+1)} < 2$  for each pair of rings
- $N_{cl} = 1$ , one cluster (only) in the LAV station

During the LAV-1 test beam in October 2009 at CERN, runs with 2 GeV electrons have been taken. In these runs, a fraction of events are generated by muons from the beam halo. The total TDC hit multiplicity is shown in Figure 8. The separation of electrons, whose hit mean value is  $\approx 15$ , and muons, with TDC hits  $\leq 7$ , gives a first indication of the capability of distinguishing muons from electrons or photons in a single station by using a simple logic.

**High multiplicity trigger:** to identify EM-showering particles

- $N_{tot} > 15$  OR  $E_{tot} > 20$  GeV
- $E_{ring} > 2.5$  GeV  $\times N_{ring}$  for at least 2 rings
- $N_{cl} > 2$ , more than 2 clusters in the LAV station

The trigger can identify the presence of high energy into one or more LAV stations indicating that the event will contain one or more electrons or photons. It can be used by the L0 central processor to identify  $\pi^+ \pi^0$  or  $\pi^+ \pi^0 \pi^0$  events having one or more photons into the LAV.

Once the trigger primitives are produced by the L0 FPGA by the TEL62 of the involved station, the board sends this information, together with the event time stamp, to a L0 LAV concentrator using a dedicated Gbit Ethernet interface. In the LAV L0 concentrator the information coming from the 12 stations are combined together (see Figure 9). The aim of the concentrator is to try the association of clusters in different stations coming from the same particle as well as to determine the particle multiplicity in the whole LAV: muons traversing more than one station, showers starting in one station generating an energy leakage in the following ones, or more than one photon/electron/muon firing the LAV stations.

All time and charge information coming from the TEL62 of each station are first aligned in time, to correct for the time of flight, knowing the station position along the blue tube. For all clusters, the times are compared to check the hypothesis that they come from the same decay, and counted. Then, for each cluster the azimuthal position is computed and compared with that of the following station. If clusters match in azimuth, they are merged: the energy is summed, while the time of the particle is computed averaging hit times.

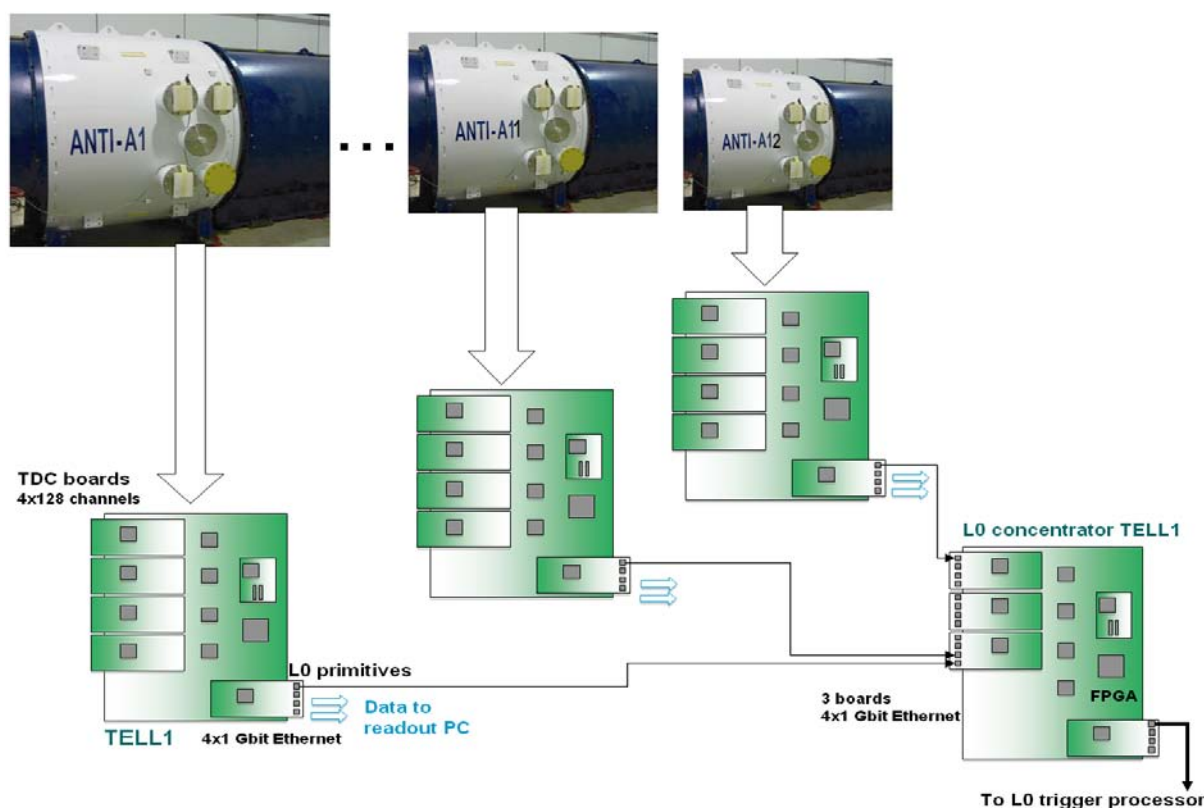


Figure 9: LAV L0 primitive generation scheme.

As results of the computation, a LAV structure is filled for each particle, with the following information:

- Particle time (at the position of a given station, e.g. LAV 12)
- Particle total energy deposit
- Total energy deposit in each of the crossed stations
- Azimuthal position (at LAV 12 position)
- Total number of crossed stations
- Total number of hits in each crossed station

Using the above information, better muon identification can be achieved exploiting the number of crossed stations and the energy deposit in each station. In fact, a pion can cross a single LAV station without producing a shower, but the probability of no conversion becomes lower and lower when the number of crossed stations increase.



Once the information are collected, the L0 primitive information encoded in two 32-bit words will be sent to the L0 Trigger Processor using a dedicated GbE link, thus saturating at 16 MHz. Assuming a muon-dominated rate of about 4 MHz (LAV OR), this leaves a safety factor of about 4. The 8 reserved bits can be used to encode the total event charge if needed by L0 central processor. The remaining particle information are stored in a circular memory buffer and sent if required to the L1 central processor.

In the present scheme the concentrator function can be obtained using a TEL62 and 12 Gbit Ethernet links: *e.g.* the 12 Gbit (double-slot) optical receiver mezzanine card developed by the LHCb experiment for the TELL1 (O-Rx card (10)). The computing power required for the data processing can be provided by the SL FPGA on-board the TEL62. The processed data are sent to the L0 central processor by one of the 4 standard TELL1 Gbit Ethernet connections. Compared to a daisy-chain architecture this scheme uses only 1 of the 4 Gbit Ethernet links housed by standard TELL1 and does not need the implementation of inter TELL1 communication protocol.

### 1.1.10 RICH System

The RICH detector is used in the trigger and offline to enhance the selection of events with a charged pion. The excellent time resolution of the RICH (100 ps) can be exploited in the L0 trigger to determine the reference time of the tracks. At L1 trigger level the RICH will provide the number and the position of the Cherenkov rings, helping to reject events with more than one Cherenkov ring, and at L2 trigger level such information can be combined with the spectrometer information in order to select pion candidates. As described in the corresponding chapter, the RICH detector has two active regions covered with approximately 1000 photomultipliers each, where the Cherenkov light produced by charged particles is focused by a spherical mirror mosaic. After the preamplifier electronics the 2000 signals are sent to 64 boards equipped with 4 NINO chips each (each NINO chip handles 8 input signals), acting as preamplifiers and discriminators. The LVDS output signals, with a time duration proportional to the time over the NINO threshold, are sent to the readout boards by means of 32-pair shielded cables. The cables and the corresponding connectors were chosen to preserve the excellent rise time of the output signal produce by the NINO chips, yielding low jitter time measurement.

#### Readout

The common NA62 TDC readout system is used for the RICH. Taking into account the requirements for the offline time resolution of the RICH (time resolution for a track better than 100 ps in the momentum range between 15 and 35 GeV/c), the LVDS differential signals, coming from the Front-End electronics (NINO boards), will be digitized by the HPTDCs on the TDCB cards described in section 1.1.5. Since a TDC card can handle 128 channels, two complete TEL62 boards will be used for each active region, for a total of 4 fully-equipped TEL62 boards (2048 channels), allowing for 2% of spare channels. The four TEL62 will be housed in the same 9U crate, placed close to the RICH detector.

The default TDC controller FPGA firmware will be used. In order to generate L0 trigger primitives as described below, a pre-processing algorithm can be applied before the transmission of the data in the PP-FPGA, in order to reduce the transit time of the data through the building stages of the trigger.

The TEL62 firmware for the RICH (both for the PP-FPGAs and the SL-FPGA), are modeled on the common default TDC firmware framework, as the main functionality of the TEL62 will be the same for all the detectors exploiting such board. The data will be stored in large memory buffers (more than 2GB for each PP-FPGA) waiting for the trigger. The memory is organized in pages; each page (corresponding to hits within a 25 ns wide time window) is addressed by synchronized timestamp counters in the TEL62 for writing. In case a positive L0 trigger is received, a programmable number of pages around the trigger will be readout from the memory being addressed by the corresponding trigger timestamp value.

During data taking, the monitoring of the board functionalities will be an important activity: in particular the buffer occupancies and other relevant working parameters will be directly evaluated in the FPGAs and transferred to the readout PC's, to spot malfunctioning, both in the readout system and in the frontend electronics. In case of special trigger requests (EOB<sup>12</sup>, test triggers, etc.) the FPGA control trigger FSM (Finite State Machine) will provide statistics and reports collected by the internal monitoring blocks.

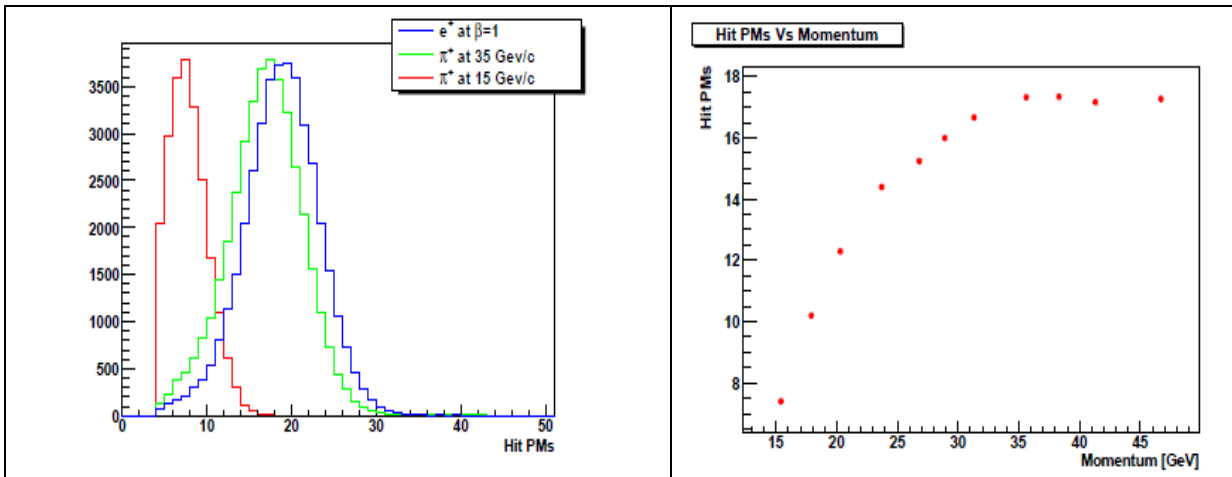


Figure 10: Left: number of firing PMs for pions and electrons.  
Right: average number of firing PMTs for pions as a function of pion momentum.

The FPGAs in the RICH TEL62s will be also used to evaluate L0 trigger primitives, in order to contribute to the L0 trigger decision. In this respect both the PP-FPGA and the SL-FPGA contain dedicated blocks specifically designed for the RICH.

The data from the TEL62 will be directly sent to the L1 PCs, through commercial Gigabit Ethernet switches. The time-aligned data from different parts of the RICH will be merged directly in the L1 PCs.

The charged particle rate within the geometrical acceptance of the RICH detector is  $\approx 11$  MHz (10 MHz from kaon decays and 1 MHz from muon halo). With 8 bytes of data for each hit (channel number, leading and trailing edges' times), and assuming to have 20 firing phototubes for each particle due to

<sup>12</sup> EOB  $\equiv$  End Of Burst

Cerenkov light (the average number measured during the 2007 and 2009 test runs for ultra-relativistic particles being about 18, see Figure 10), a readout bandwidth of  $\approx 1.8$  GB/s is needed for the entire RICH. If the phototubes will be adequately mapped onto the TDCs in order to balance their average load, each TDC board will transfer to the corresponding PP-FPGA approximately 110 MB/s of data.

The I/O speed of the TEL62 DDR2 RAM memory is greater than 1.5 GB/s. This bandwidth is more than sufficient for both write and read operations. The buffer space available to store data on the original TELL1 board is 96MB, already enough for 0.9 s of data taking, so that the maximum overall L0 trigger latency for L0 is not an issue. In the new TEL62 the memory will be substituted with a faster and bigger DDR2 SDRAM, allowing bigger safety factors.

Assuming a L0 trigger rate of 1 MHz, the TEL62 output bandwidth will be  $\approx 50$  MB/s, which should be sustainable by one Gigabit Ethernet link (max 125 MB/s per channel).

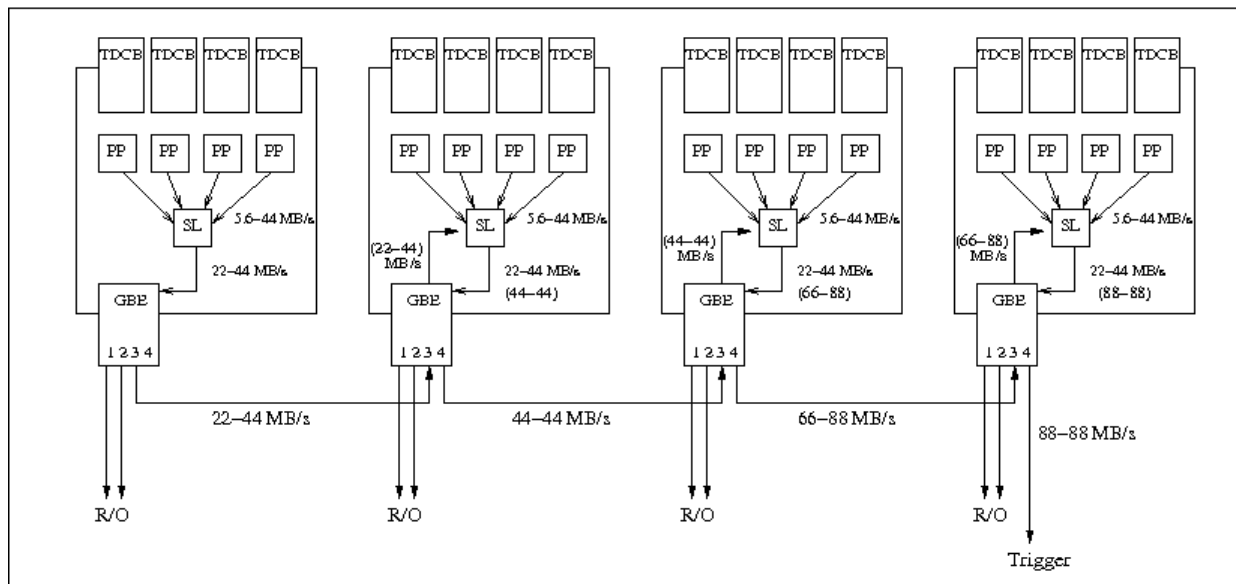


Figure 11: TEL62 daisy-chain connection for L0 primitive information. Bandwidths are indicated for “best” and “worst” cases; total bandwidths in a given stage are indicated in parenthesis.

## L0 trigger

The RICH can contribute to the L0 trigger decision by providing a precise time reference for the charged particle. In this case the RICH L0 trigger primitives will be generated directly within the TEL62, by counting the multiplicity of the hits within fine time bins. For each memory page (25 ns wide) stored in the main buffers, the hits are further subdivided in 8 time windows (3.125 ns wide) in a histogram-like fashion. A threshold condition on the maximum number of single bin entries (time coincidences), can be applied in order to define the trigger primitives. A final time histogram for the whole RICH detector has to be constructed in three different steps: the PP-FPGAs (receiving data from 128 channels) define the first partial histograms; four of these are merged together in the SL-FPGA at TEL62 level. Finally the four histograms, produced by each TEL62 have to be merged again to obtain the final

result, from which a decision on the basis of the overall RICH multiplicity can be taken. The last merging can be performed by either connecting the four TEL62 boards in daisy-chain, using for this purpose two of the four GbE links on each board (see Figure 11), or by using an additional (fifth) TEL62 board equipped with Gigabit Ethernet mezzanine receivers (in this case only one GbE link is used for this purpose on each of the four sender boards).

In order to build a flexible trigger algorithm, the time histograms will also allow storing information from hits that were not previously ordered in time. In such a case the expected maximum time disorder, related to the time required for a hit to reach the PP-FPGA from the TDC, defines the minimum time interval for which each histogram must be left “open” for writing, in order to accept further entries.

The minimum reasonable size for each L0 trigger primitive packet is 8 bytes, including timestamp (4 bytes) and the bin contents of the histogram (4 bits x 8 time bins). In the “best case” all the hits from the same event appear on the same PP-FPGA, resulting in a histogram rate of 0.7 MHz per PP-FPGA (= 11 MHz/16), corresponding to 5.6 MB/s from each PP-FPGA to the SL-FPGA. The corresponding SL-FPGA outflow will be 22 MB/s in this case. In the “worst case” the hits from one event are spread over 8 PP-FPGAs maximum (those handling one RICH spot), resulting in a bandwidth of 44 MB/s from the PP-FPGA to the SL-FPGA. In this case the SL-FPGA has to merge the histograms belonging to the same event, and its outflow is 44 MB/s. Using the same logic, in the daisy chain architecture, the last Gigabit Ethernet link would be loaded with 88 MB/s both in the best and worst cases. For the proposed scheme a single GbE link can be used to bring 8 bytes trigger primitives to the L0TP.

To optimize the throughput from the different FPGAs, several geometrical configurations for connecting PMTs to TDCs were considered. Simulation results (14) show that there is no connection configuration in which it is possible to obtain significant reductions in data transfer rate without a significant efficiency loss. In order to maximize the rate of data sent from the TEL62 to the L0 Trigger Processor, L0 trigger primitive words corresponding to several events will be merged in a unique Ethernet frame (including few control words) exploiting jumbo frame support.

To provide L0 primitive time histograms, the SL-FPGA firmware will have to be modified with respect to the default, to include the logic for the histogram merging stage and – in case a daisy-chain connection configuration is chosen – also that to handle the Ethernet links also as receivers for the primitive data from the other boards.

A further use of the RICH in the L0 trigger is given by the possibility to have the analog OR of 8 channels being discriminated by the same NINO chip (precise hardware time alignment of the different channels is mandatory in this case). The set of 250 OR signals could be digitized in two TDC boards (one for each RICH spot) and following the scheme already described, a RICH multiplicity information could be extracted within a single TEL62. The cost for the generation of L0 trigger primitives with this scheme is that a dedicated TEL62 must be employed in addition to the four ones used for readout. Moreover, the signals corresponding to the hardware OR produced by the NINO chips must be recorded, in order to preserve the possibility to have full offline reproducibility of the trigger algorithms.

Finally, additional trigger information from the RICH would also be useful at L0 to generate control triggers or special triggers for other physics studies (*e.g.* triggers on high multiplicity of charged particles).

## L1 trigger

The data selected by the L0 trigger will be processed by the L1 PCs in order to apply more sophisticated selection criteria at the single-detector level. Four GbE links with 50 MB/s of data could be handled, in principle, by a single PC equipped with a quad-port Gigabit card (*e.g.* INTEL PRO/1000 GT). An internal PC RAM of 4 GB is enough to store the whole burst, leaving space for buffering the results. The maximum L1 allowed latency (of the order of a few  $\mu$ s) would have to be shared between the time used by the L1-RICH PC to compute the trigger primitives, and the time used by the L1 Trigger Processor PC to take the final decision on the event. Assuming a total spill duration of 10 s and a L1 trigger input rate of 1 MHz, the maximum computing time is 1  $\mu$ s times the number of L1 processes running in parallel (corresponding to the number of L1 CPUs). At L1 the fundamental information to be computed for the RICH is the number of Cerenkov rings in the event (corresponding to the number of charged particles) and the geometrical parameters related to such rings. Preliminary tests show that a ring fit based on a maximum-likelihood approach takes more than 200  $\mu$ s of computing time per ring on an Intel i7-950 CPU (3.07 GHz, 12 GB RAM), requiring at least 200 CPU cores to be used in order to allow a real-time processing of the data selected by the L0 trigger during the spill (assuming to have events with only one track, which is moreover not the case). Some gain in processing time can be obtained by either improving the fitting procedure or by decreasing the resolution requirements. In any case a multi-core L1 PC cluster must be considered for this purpose. The required size of the PC cluster will be determined after a more complete queue simulation, including other parameters like the bandwidth available from L1 to L2 and the processing capability of the L2 processing farm. Nevertheless, a solution based on the use of GPUs (video processor cores) is under investigation: such an approach dramatically increases the computing power of a single PC.

The following variables will be computed and used to apply selection cuts for the main decay mode and for ancillary triggers:

- N\_ring : number of independent reconstructed rings;
- N\_hits[i]: number of hits in the i-th ring;
- R\_ring[i]: i-th ring radius;
- T\_ring[i]: time of the charged particle associated to the i-th ring (average time of the hits participating to the fit);
- Chi2\_ring[i]: Quality of i-th ring fit;
- X\_ring[i], Y\_ring[i]: position coordinates (x,y) of the center of i-th ring.

These quantities could be used to define L1 trigger selection criteria based on track multiplicity, ring quality and ring time, in order to reject events with a large number of reconstructed rings or events with a bad quality of the performed fits. At this stage cuts on the spatial distribution of the hits belonging to the same ring could also be applied, for instance to remove rings that are only partially contained on the RICH acceptance (arcs).

### 1.1.11 Charged Hodoscope (CHOD) System

The existing charged hodoscope (CHOD) from NA48 will be re-used in NA62. It is placed after the RICH detector to reject photo-nuclear interactions with the RICH material. Depending on the readout architecture the CHOD could also be employed -as backup for the RICH- to define the reference time of all the triggers with at least one track in the final state, (or to measure the RICH trigger efficiency). The CHOD would have the advantage that all channels are handled from a single TEL62 board.

The CHOD readout and trigger architecture could use the common NA62 building blocks. The light generated in plastic scintillator slabs are collected from photomultipliers. Thanks to the similarity of the CHOD signals with respect to those of the RICH, the analog signals will be discriminated in the same front-end described in the RICH chapter, exploiting the 32-channel preamplifier/discriminator boards housing 4 NINO chips each.

### **Readout**

The readout of the digital signals will be based on the common NA62 TDC system described in section 1.1.5. The CHOD has 128 channels, which are arranged in two planes and they could be read out by two TDC boards<sup>13</sup> leaving also some spare channels. For an 11 MHz maximum rate in the CHOD geometrical acceptance, the output rate of each TDC board will be, on average, less than 40 MB/s, which can easily be managed by the corresponding PP-FPGA. The particle hit rate isn't uniform over the hodoscope surface, and some care must be taken in connecting the CHOD channels to TDCs, in order to avoid excessive rates on a single TDC board. In any case, using a single TEL62, the readout bandwidth will be around 100 MB/s, which can be handled by 1 or 2 Gigabit Ethernet links.

Due to the length and the age of the scintillators (slabs of up to 1.21m) the intrinsic time resolution of a single detector plane remains limited to a range between 3 and 5 ns. The offline time resolution using both planes was measured to be 200 ps (13).

### **L0 trigger**

If the CHOD is used as time reference for the L0 trigger the online time resolution should be at the level of 1 ns. In this case the time information from both detector planes has to be used online. Slewing and propagation time corrections should be applied directly in the PP-FPGAs (assuming enough information is available there).

The L0 trigger primitive data are sent to the L0 Trigger Processor in 8-byte frames including detector ID, timestamp and "fine time". To exploit the time resolution of the CHOD ( $\approx 1$  ns online), the 8 bits reserved to the "fine time" will encode the following information:

- the 5 most significant bits will identify the fine time window (within the 25ns period defined by the timestamp) where the hit(s) occurs with a resolution of  $25/32$  ns (0.78 ns);
- the 3 less significant bits will contain the hit multiplicity within the fine time window, (*e.g.* 1hit, 2hits, 3 hits,  $\geq 4$  hits).

If one or more hits (even if generated by the same decay) happen to occur in adjacent fine time windows, a second frame is generated and sent to the L0 Trigger Processor.

For a rate of 11 MHz in the CHOD acceptance, the L0 primitive data rate from the TEL62 to the L0 Trigger Processor will be 88 MB/s, and such bandwidth could be handled by one of the four available GbE links. In order to reduce the amount of the data to be sent, several frames could be merged into a single packet, allowing including the detector ID only one time (latency issues should be taken into account in this merging, of course).

---

<sup>13</sup> Ultimately even one TDC board would be sufficient.

Since the online time resolution will be fundamental to trigger on the signal events with good efficiency (this is not trivial to be obtained, given the presence of veto conditions in the L0 trigger algorithm), the time of all counters must be aligned with good precision. Besides the intrinsic offset correction capability of the TDC themselves, this aim can be reached including on the PP-FPGA a look-up table with pre-calculated offsets to be subtracted from the incoming times measured by the TDCs.

The opportunity to measure the L0 trigger time with a single additional TEL62 reading all the counters of the charged hodoscope and the 250 analog OR signals available from the RICH is under investigation. In this case the information from two independent detectors could be merged at L0, contributing to a better online definition of the L0 trigger time.

### L1 processing and trigger

Once the data collected by the CHOD will reach the L1 PCs, both the time and the multiplicity of the events can be re-evaluated in software, and given the better resolution on the time measurement ensured by the presence of the complete information, tighter cuts could be applied at this level. For instance, by exploiting the available information on the time-over-threshold, a slewing correction could be applied to the hit time measured by the TDCs, to correct for the bias introduced by the fixed threshold of the discriminator chips.

The size of the PC farm required will be smaller with respect to the RICH case, due to the simpler reconstruction algorithms.

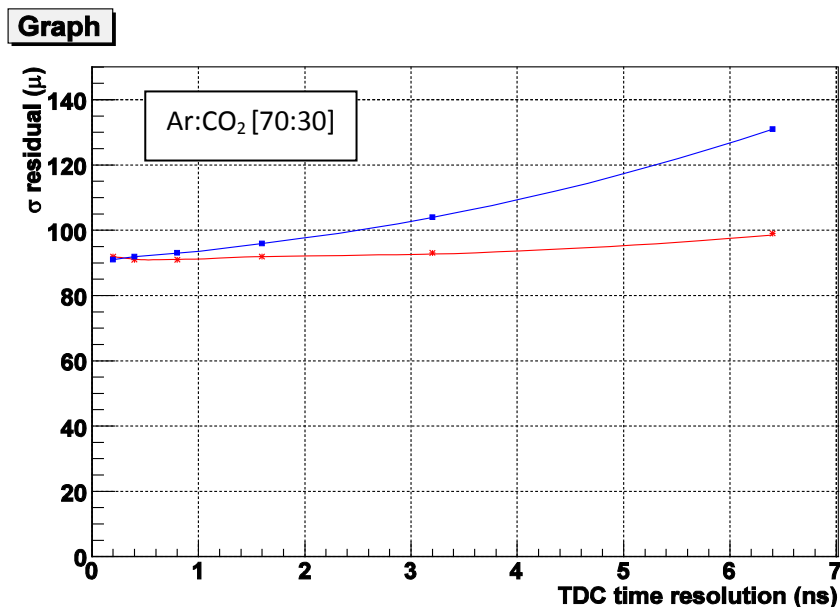


Figure 12: Straw resolution versus TDC time binning (blue: unknown space-time dependence; red: known space-time dependence).

### 1.1.12 STRAW System

The Straw readout electronics should provide track data to the NA62 DAQ system in required format, perform online monitoring to check data quality, control the front-end electronics, and possibly trigger on a single charged particle or veto on multiple charged tracks in an event. For data extraction, front-end control and online monitoring a straw detector specific module is used: the Straw Readout Board (SRB). For data collection and event building, handling and selection we will use the readout board (15), which is common to the majority of detectors in NA62.

#### Time to digital conversion

To understand the required drift time resolution, a dedicated study has been performed with different TDC time bin steps. The targeted position resolution was specified to 130  $\mu\text{m}$  (see Straw Tracker chapter). The result indicates (see Figure 12) that for a straw with known position-time dependence, even a 6 ns time bin of TDC (time to digital converter) would be sufficient, when using an Ar/CO<sub>2</sub> gas mixture.

However, other constraints led to fix the TDC time-binning to a maximum of 3 ns; these include mainly the matching to the GTK, where a time binning of 6ns in the straw detector would require opening a too large window for track fitting. On the lower side, using a very small TDC bin is useless as the space resolution is dominated by multiple scattering of particles (see Straw Tracker chapter). This is the reason why the use of the common TDC system developed for NA62, with much higher time resolution is not needed.

We plan to implement the TDC directly in a FPGA (Field Programmable Gate Array), together with other readout functions. A preliminary study shows that one can achieve 1 ns resolution (1.6 ns bin) with a cost-effective version of FPGA.

#### Readout

The readout architecture follows the detector partitioning. The smallest readout unit is a module of 16 straws served by 1 front-end board. 30 such boards form a view, and each chamber has 4 views (x,y,u,v). The whole detector has 4 chambers. The data from 15 front-end boards (half a view) is collected by one SRB, which also provides a control for thresholds and test pulses. The chamber (4 views) is thus served by 8 SRBs housed in one VME 9U crate, positioned about 5 meters from the detector. As 2 chambers are within a short distance, it is possible to use a single full VME 9U crate (21 slots) for housing SRBs for two chambers (16 boards). SRBs will receive precise system clock, timing information and control from the common NA62 TTC system, and will time-align the data from the straws by attaching the required timestamps. The data sent from the SRBs to the TEL62 boards will have a fixed format of 40 bits per hit:

Timestamp 30 bit	Fine time 4 bit	Leading/trailing edge 1 bit	Straw ID 4 bit	Control 1 bit
---------------------	--------------------	--------------------------------	-------------------	------------------

This information will be re-arranged within the TEL62 to provide the final readout data.

For the data extraction and positioning of TDC two approaches are considered (Figure 13).



- (1) TDCs are placed on the SRB. For data extraction, control and services for the front-end board 5 meters long halogen-free SCSI cables are used. As these cables are rather heavy and rigid, appropriate attention must be paid to fixation and support, mainly close to the front-end board connection.
- (2) TDCs are placed directly on the front-end board. For data extraction, control and services standard halogen-free Ethernet cables are used. Such a solution is mechanically more stable and does not provide a big strain on the PCB connector. Also, such a solution would be advantageous from a noise point of view, as the signal spectra from high-speed serial links lies outside the sensitive bandwidth of front-end electronics. In this case, a SRB collects data over a high-speed serial link running at 320Mbits/s (or more). If needed, for boards close to the beam with high data rate, two of them can be used. There is one dedicated line for clock and one for the control. The control line can use Manchester encoding as the number of bits to transmit is very low; data lines would use 8b/10b encoding to ensure a return-to-zero bit stream.

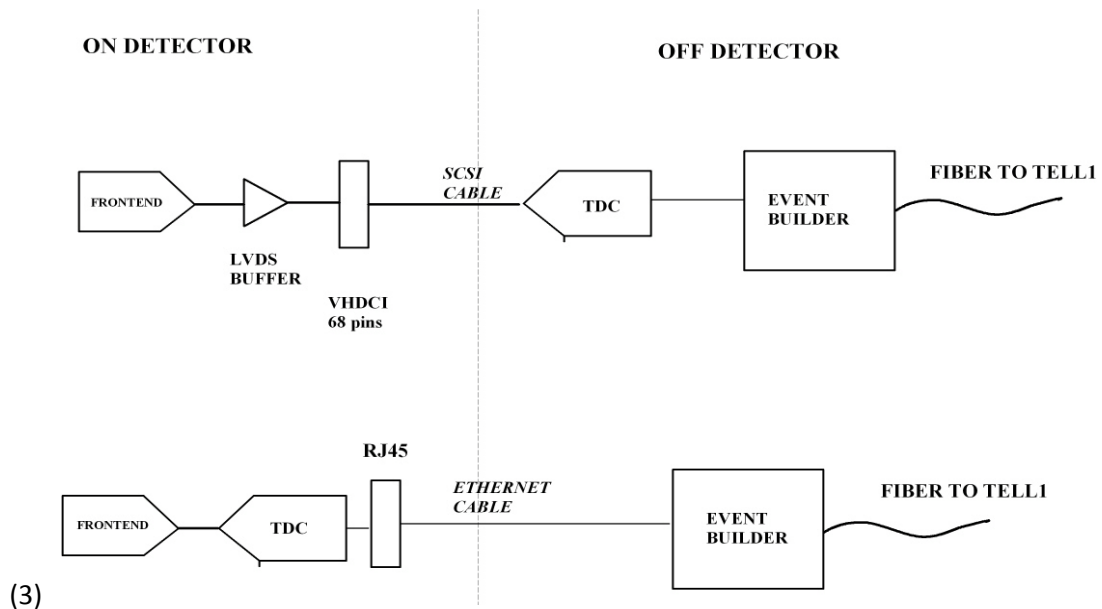


Figure 13: Two possible architectures for the Straw TDC readout.

All 16 SRBs from one VME crate transmit formatted data to one TEL62 board, so for the whole detector two TEL62 boards are required. For the data transmission from SRB to TEL62 either copper or optical links might be used (see Figure 14).

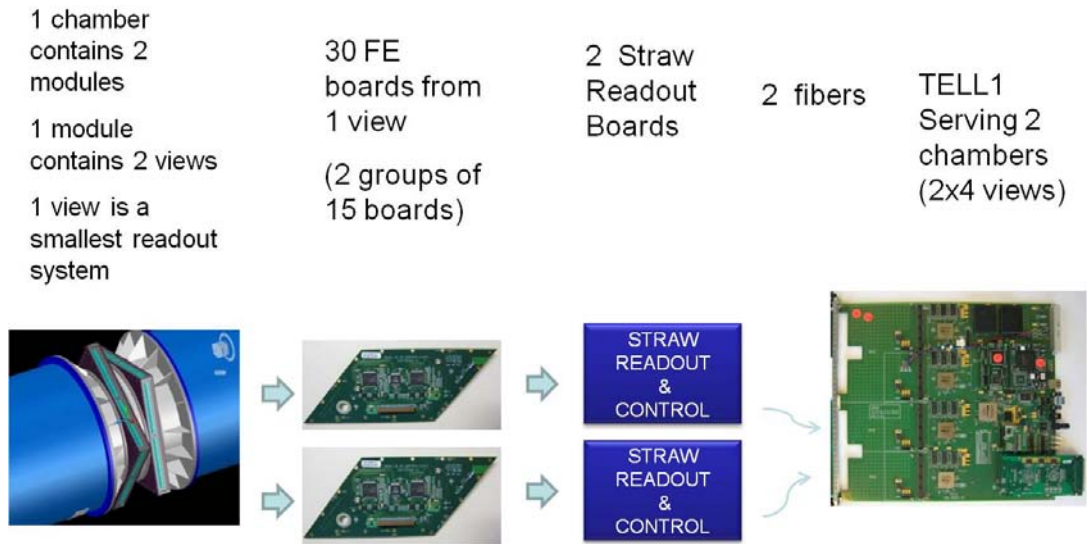


Figure 14: Straw readout scheme for one View (1/2 a Chamber).

Each TEL62 board collects data from 16 SRBs. If optical links are used, the double-width mezzanine board developed for the TEL62 by the LHCb collaboration with 2x12 optical links can be used. If copper links are used, a dedicated TEL62 mezzanine receiver board will have to be developed for this purpose. The firmware on the TEL62 and its control software will be Straw detector specific: its main tasks will be to check the integrity of incoming data, building events from matched timestamps and event management; in case the STRAW detector participates to the L0 trigger formation, the evaluation of trigger primitives would also be handled there.

Upon arrival of the L0 trigger, it sends the requested event with the matched timestamp to the DAQ system. The data sent from the TEL62 board corresponds to hits in straws from two chambers, *i.e.* eight views. When all straws along the particle track generate a signal, the amount of data sent per event is:

$$40 \text{ [bits]} * 4 \text{ [views]} * 2 \text{ [chambers]} * 4 \text{ [straws in view]} * 2 \text{ [lead + trail edge]} = \\ = 2688 \text{ bits/event}$$

If neither of the above described approaches works, a fallback solution is identified using the common system of TEL62 boards with mezzanine TDCs. This solution will require a board between front-end and TEL62 for rearranging data lines, providing front-end control, distributing power supply and enabling online and DCS monitoring. The first estimate gives up to four such boards per view, 64 boards for the whole detector; most probably in 9U VME format using VME interface. A TEL62 board with four mezzanine TDCs can handle 512 channels. As one view of a chamber contains 30 front-end boards with 16 channels, one TEL62 board per chamber view is needed, and 16 TEL62 boards for the whole detector.

#### Expected data rates

The average rate of particles per straw is estimated to be 33 kHz. We expect to have 1 signal from each particle, of which both trailing and leading edges times have to be measured. Thus the average data rate per straw is 40 bits x 2 x 33 kHz = 2.6 Mbit/s, or 42 Mbit/s per front-end board serving 16 straws.

There are few straws close to the beam with much higher particle rates, up to 500 kHz. The maximum data rate will be 40 Mbit/s for such straws. To this data rate one should add extra data caused by noise, spurious hits, etc., which must be kept as low as possible.

The data rate from one TEL62 board to the PCs depends on the L0 trigger rate, *e.g.* for 100 kHz the data rate is 270 Mbit/s. For such a rate, one gigabit Ethernet link would be sufficient, but for the design value of 1MHz L0 trigger rate 3 links are needed. If TEL62 boards could not cope with this data rate, it is possible to add data buffer on SRB boards and transmit data from SRBs to TELL1 only on L0. In that case, building and event management would be pushed back to SRB.

### **Clock distribution, control and timing of the readout electronics**

The Straw detector will use standard NA62 TTC modules, LTU and TTCex, for control and clock distribution. One set of modules is sufficient for the whole detector as there are two VME crates and one TEL62 crate as destinations. TEL62 boards will use the full TTC protocol, while SRB boards need only the clock and “Start Of Burst” signal for synchronization. SRB boards will provide fine tuning of clock delays for timing of the detector. Adjustable delays are needed due to the spread of propagation time through components and cables, and time-of-flight of particles along the beam.

### **Trigger and vetoing**

Both leading and trailing edges from the straws’ signals provide useful information. While from the leading edge time one can obtain the precise crossing position of the particle through the straw, the trailing edges occur at the same time for all straws hit by the same particle, independently from their crossing distance from the wire. The trailing edge time is used as a validation of straws on a track, thus reducing false hits and improving track fitting.

It is also rather straightforward to use the trailing edges times for a fast hardware trigger or veto. For this purpose all the views in the chamber are partitioned into corridors, the size of the partition depending on trigger granularity. Each view has 4 straw planes and the geometry of straws placement guarantees that at least 2 straws are hit by each passing particle. Thus one should expect to have from a minimum of 2 up to 4 straws with signals having the same trailing edge time (see Figure 15). A View Trigger Logic (VTL) opens a time window, which should cover the spread of trailing edges from one view, and if there are at least 2 straws in the window with the same trailing edge it sends the corridor number and timestamp to the Chamber Trigger Logic (CTL). The CTL collects numbers from all views and opens a timing window of the order of 7 ns to compensate for signal propagation time in straws from different views. If there are hits from different views, it sends coordinates and timestamp to the central trigger. Using coordinates and timestamp, a corresponding particle could be searched for in other sub-detectors. Such information could be used for generating: (a) a trigger, when there is only one track; (b) a veto, when there is more than 1 track.

## Corridor example

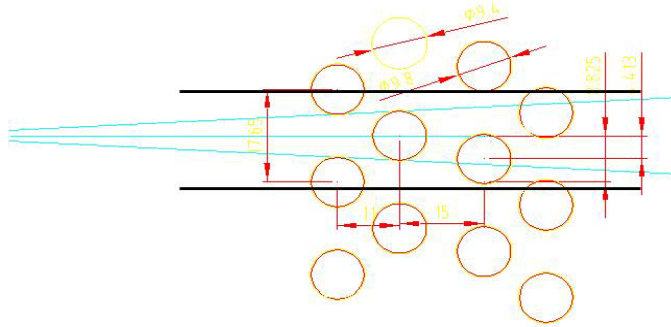


Figure 15: Smallest possible track corridor in 1 view.

### 1.1.13 MUV System

In total (with MUV3 option A), 560 channels have to be read-out (Table 13) and it is foreseen to use the standard NA62 read-out approaches. In this respect, for MUV1 and MUV2 stations two data acquisition choices exist:

1. **The “LAV” readout:** after suitable discrimination, the PMT signals are routed to TDC boards, housed on the TEL62 boards. The signal amplitude is determined by measuring the time-over-threshold in the TDCs. To cover the dynamic range of the analogue signals, two digital lines are used, with different thresholds. The corrected time and signal amplitude are reconstructed from four measured times (two leading and two trailing) per PMT. With 128 channels per TDC board and a maximum of 4 TDC boards per TEL62 module, one or two TEL62 boards could be used to read MUV1 and MUV2.
2. **The “LKr” readout:** the CREAM readout of the LKr calorimeter is based on flash ADCs with 40 MHz sampling, 14 bit resolution, large data buffering and optional zero suppression. To be able to interface the CREAM read-out directly the MUV1+2 signals will have to be shaped to a comparable timing than those of the LKr, *i.e.* 20 ns rise-time and 2.7  $\mu$ s fall-time with maximum amplitude of 1 V. The signals are then re-shaped, so that the FADC receives a signal with 40ns rise-time and 70ns FWHM. One additional VME crate with 6 (3) CREAM modules would be adequate to acquire the data from MUV1 (MUV2)

The MUV3 system is used as fast muon veto, mostly for the L0 trigger, and only timing information is needed. The signals are discriminated with existing Constant Fraction Discriminators<sup>14</sup> compensating jitter from amplitude fluctuations. The time measurements are done with the standard TEL62/HPTDC readout boards. One board with 512 channels can receive the data from the full MUV3 station.

<sup>14</sup> From the NA48 AKL detectors.

Table 13 Number of read-out channels of the MUV detector.

Module	Number of Channels
MUV1	176
MUV2	88
MUV3 Design A (B)	296 (252)
<b>Total</b>	<b>560 (516)</b>

### 1.1.14 Charged ANTI (CHANTI) System

The conceptual scheme of the CHANTI readout is shown in Figure 16.

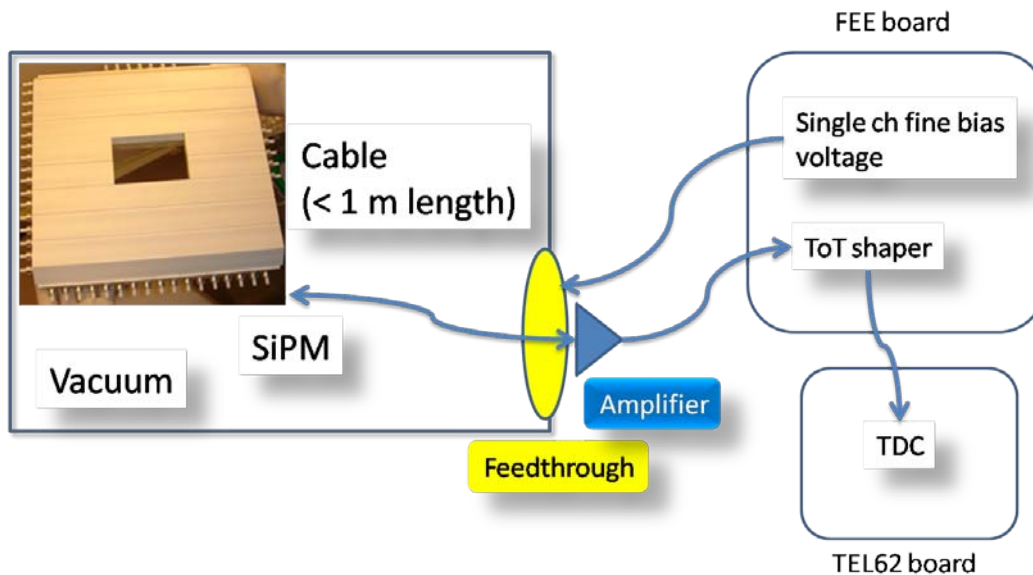


Figure 16 Schematic view of the CHANTI readout.

Each CHANTI station is composed by 46 scintillator bars, for a total number of 276 channels. After amplification, signals are processed by FEE boards similar to the one used for the LAV system.

The 9 FEE boards (32 channels each), will have to provide, for each channel:

- a way to control the  $V_{\text{bias}}$  with  $O(10 \text{ mV})$  accuracy
- a fast, DC coupled, conversion to a Time Over Threshold-LVDS signal output
- a temperature and/or a dark current monitor for slow control adjustment of the  $V_{\text{bias}}$

Thresholds and  $V_{\text{bias}}$  settings will be controlled using the CANOpen standard. Since the dynamical range of the signals is expected to be small (the detector is essentially sensitive to MIPS) a single threshold will be used.

The LVDS output will be directly sent to a standard TEL62 system equipped with TDC boards for both leading and trailing edge measurement. One TEL62 board equipped with three TDC boards will be enough for the entire detector, allowing also for a good number of spare channels.

## **L1 trigger**

The CHANTI is not providing primitives for the L0 trigger.

An evaluation of the maximum L0 data rate can be done as follows. The highest multiplicity in the CHANTI is expected from the beam halo events crossing all six stations. If fully efficient on these events, the system will give at most  $4 \times 6 = 24$  hits or 192 bytes per event. These events occur at 1 MHz rate. Inelastic interactions in the GTK will be detected at approximately the same rate, with comparable hit multiplicity. One can safely estimate a maximum of 200 bytes at 2 MHz. Assuming a 1 MHz L0 trigger rate and a readout time window of  $O(100 \text{ ns})$  one expects  $O(200 \text{ kHz})$  of such events in coincidence with a trigger. This would generate a data rate of about 40 MB/s well below the TEL62 specifications.

After zero suppression done at TEL62 level, the L1 farm checks data integrity and performs further data reduction. Raw times will be corrected for individual channel calibration constants and corrected for time propagation along fibres by using x-y coincidence algorithms.

Moreover, time slewing corrections could be implemented by exploiting the ToT-amplitude correlation. This will allow a tighter time window cut, which will be optimized according to the maximum allowed L1 rate. If necessary beam halo events could be recognized at this level by means of appropriate multiplicity and collinearity algorithms.

One multi-core PC farm should be envisaged for L1 processing. If compared to the RICH system, given the less demanding L1 algorithm complexity and the 1 MHz expected L0 rate, the total number of CPU cores will be considerably less.

### **1.1.15 LKr (LKr) Readout system**

The 13'248 channels of the LKr electromagnetic calorimeter are continuously digitized with 40 MHz flash ADCs. The calorimeter is therefore the largest single data producer in NA62, generating 800 GB/s of raw data.

The initial current is derived from the charge measured by a preamplifier mounted inside the cryostat at liquid Kr temperature and connected to the anode electrode by a blocking capacitor. The integration time constant chosen for the charge preamplifier is 150 ns. The signal from the preamplifier is transmitted to a combined receiver and differential line driver mounted outside the calorimeter close to the signal feed-through connectors. The receiver amplifies the preamplifier signal and performs a pole-zero cancellation. Those two elements were built for the NA48 experiment and will remain untouched for NA62. The dynamic range (50 GeV of deposited energy) corresponds to an amplitude of  $\pm 1 \text{ V}$  (on a 100 ohm termination) at the input of the digitizer electronics. The required signal to noise ratio is 15000 to 1. The LKr readout chain scheme is sketched in Figure 17.

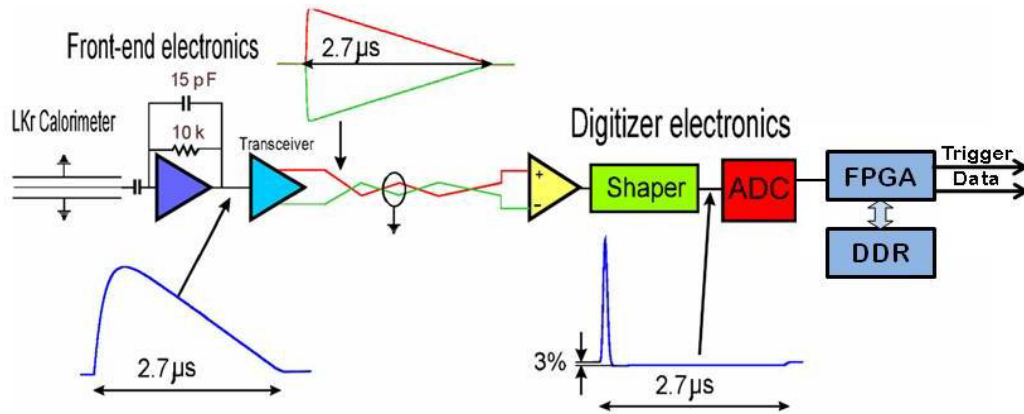


Figure 17 LKr readout scheme.

The Calorimeter REAdout Module (CREAM) is the new back-end part of the NA62 LKr data acquisition chain. CREAMs provide 40 MHz sampling of 13248 calorimeter channels, data buffering during the SPS spill, optional zero suppression, and programmable digital trigger sums for the LKr L0 trigger system. The digitization has 14-bit resolution, and the event readout rate is up to 1 MHz.

The Timing, Trigger, and Control (TTC) system (1) developed for the LHC experiments is used by all NA62 sub-detectors. The LKr-specific TTC and CREAM modules are single-width 6U VME64x units. One TTC module and 16 CREAM modules are housed in one VME64x crate and controlled by a VME-PCI bridge master, which is used for configuration, monitoring and test purposes. The entire LKr back-end system is composed of 28 readout crates.

### LKr clock distribution

High-speed applications using ultra-fast data converters often require an extremely clean clock signal to make sure that an external clock source does not contribute with undesired noise to the overall dynamic performance of the system.

The experiment TTC system provides distribution of 40.087 MHz clock, Level 0 trigger information, broadcasts and individually-addresses control signals. The system is used down to the CREAM crates. One LKr-TTC module interfaces TTC commands to all the boards in one crate through the VME backplane. A CERN standard TTCrq (7) mezzanine board is used to decode and convert optical inputs. This board incorporates a TTCrx ASIC chip (5) (6) (15) and a QPLL (6), which provide three programmable output clocks (40, 60/80 and 120/160 MHz) with a time jitter below 50ps. The measurements done by the ATLAS LArg group show that the TTCrx clock signal noise (related to trigger and B-channel information sent after every trigger) can be reduced below 20 ps by turning off all B-channel activity on the chip providing the ADC sampling clock. In addition, the TTCrx has an option to deskew the clock signals in step of 104 ps, which might be useful for the data alignment and calibration. For a good performance is it planned to use two TTCrx chips, one to provide a clean clock by switching off all the B-channel activity, and the other to derive the trigger information

The VME backplane P0 connector is used to deliver all TTC signals from the LKr-TTC module to all CREAMs in the crate via LVDS links.

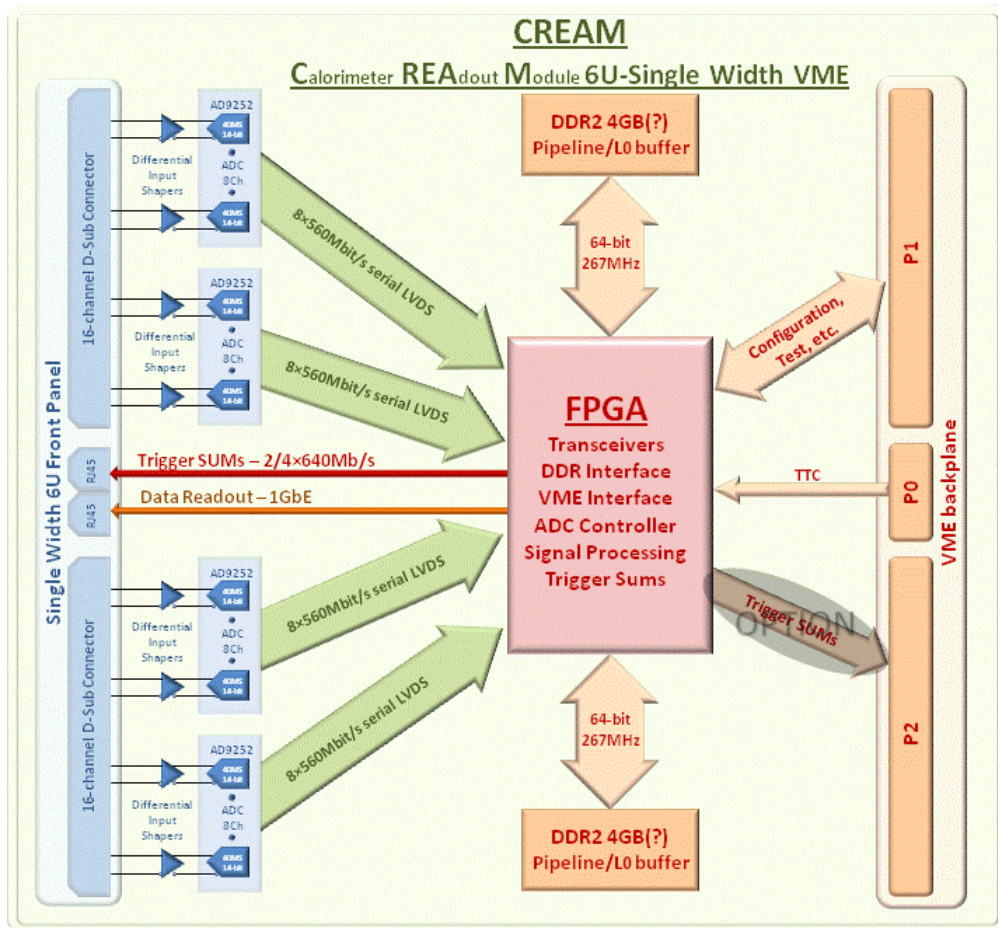


Figure 18: CREAM block diagram

### 1.1.15.1 The Calorimeter REAdout Module (CREAM)

The CREAM is a 1-slot wide 6U VME64x module. It houses 32 calorimeter channels (4 octal ADC chips).

Each ADC readout channel has a dedicated serial LVDS link to a FPGA serving 32 channels. The FPGA acquires ADC raw data at 40 MHz, performs primary treatment-formatting, and interfaces data to a DDR2 SODIMM memory module for temporary storage. The block diagram of the module is shown in Figure 18.

The VME bus (P1) is used for configuration, monitoring and test purposes. Experiment triggers and clocks are supplied via the dedicated VME backplane P0 connector. The trigger sums are sent to the trigger system via a front panel RJ45 connector. The module should be able to operate in standalone mode using on-board generated clocks and triggers, and the event memories should be writable and readable via standard VME facilities.

The CREAM data and trigger processing requirements will evolve during the experiment running, so it is expected to have upgrades of the FPGA firmware.

#### Analog input circuit



## FEATURES

8 analog-to-digital converters (ADCs) integrated into 1 package

93.5 mW ADC power per channel at 50 MSPS

SNR = 73 dB (to Nyquist)

ENOB = 12 bits

SFDR = 84 dBc (to Nyquist)

Excellent linearity

DNL =  $\pm 0.4$  LSB (typical); INL =  $\pm 1.5$  LSB (typical)

Serial LVDS (ANSI-644, default)

Low power, reduced signal option (similar to IEEE 1596.3)

Data and frame clock outputs

325 MHz, full-power analog bandwidth

2 V p-p input voltage range

1.8 V supply operation

Serial port control

Full-chip and individual-channel power-down modes

Flexible bit orientation

Built-in and custom digital test pattern generation

Programmable clock and data alignment

Programmable output resolution

Standby mode

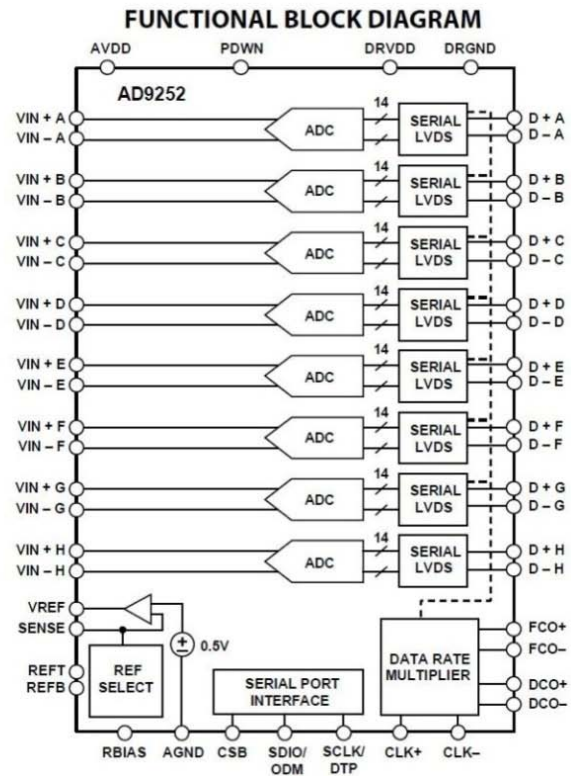


Figure 19 AD9252 ADC diagram and features.

The signal at the input of the CREAM module has a 20 ns rise-time, a 2.7  $\mu$ s fall-time and a  $\pm 1$  V maximum amplitude. Each of the 32 channels consists of a differential line receiver and a pulse shaper. The shaped ADC input is a differential semi-Gaussian signal with a 40 ns rise time and a 70 ns FWHM.

## ADC

The AD9252 ADC (16) from Analog Devices satisfies the experiment requirements. It is an octal, 14-bit, 50 MSPS ADC with an on-chip sample-and-hold circuit and one serial data link per channel. The circuit block diagram and main features are shown in Figure 19.

The ADC requires a single 1.8 V power supply and LVPECL-/CMOS-/LVDS-compatible sample rate clock for full performance operation. No external reference or driver components are required for many applications.

The ADC contains several features designed to maximize flexibility and minimize system cost, such as programmable clock and data alignment and programmable digital test pattern generation. The available digital test patterns include built-in deterministic and pseudorandom patterns, along with custom user-defined test patterns entered via the Serial Port Interface (SPI).

## FPGA controllers

Most of the CREAM functionality is implemented within two FPGAs. The first one handles the VME-bus protocol, the module configuration and provides internal communication with the other circuits on board. The “main” FPGA defines the mode of operation and configures the ADCs via SPI. It handles the ADC serial output links, reads, formats and locally stores the data in a circular buffer with a depth corresponding to at least 1ms (L0 trigger latency).

The minimum 8 data samples from each of the 14k calorimeter cells are read out from the CREAMs on every **Level 1** trigger (and not L0 as most other sub-detectors), at the rate of up to 100 kHz. Each sample consists of 14 bits, for a total of about 200 Gbit/s of data for the entire calorimeter. This data volume is demanding in term of the bandwidth of the subsequent parts of the DAQ system, as well as in term of storage capacity, therefore a zero suppression scheme is foreseen.

In parallel to the readout, the CREAM performs the summation of the selected channel samples, and sends the resulting “Super Cell” data to the LKr L0 trigger system.

Upon receipt of the L0 trigger signal, the relevant control part extracts the corresponding 8 (at least) data samples from the “pipeline” or “circular” part of the buffer and stores them in a readout buffer large enough to store up to  $1.6 \times 10^7$  64-byte data samples (corresponding to a 1MHz trigger rate during twice the 1 s latency time of the L1 trigger). Two DDR2 interfaces provide the control for two SODIMM memory modules of 2 GB each, where both linear and circular buffers are implemented. With the above figures, the size of the readout buffer is 1 GB.

A Gigabit Ethernet link is managed by the FPGA for the CREAM data readout at the L1 trigger reception, and to receive L1 triggers.

### **L0 and L1 data storage DDR2 memory**

The DDR2 memory should be virtually split into two parts. The first one serves as a pipeline (or “circular buffer”), and insures continuous data storage at 40 MHz with programmable latency (up to 1ms). The second part of the memory is used as a L0 trigger buffer. The L0 is effectively a random signal with about 1 MHz average rate, occurring a fixed time (latency) after the event took place. If a L0 command is received by the module, the corresponding data is extracted from the pipeline. A programmable number of adjacent time slices (stored in the pipeline at an offset corresponding to the L0 time) are copied from the pipeline into the second part of the memory (L0 buffer) and then read out from the module after a positive L1 trigger is received on the GbE link.

### **Trigger sums**

During the data acquisition, digitised signals from the selected channel are summed up to build a Trigger Sum (Super Cell) to be sent to the LKr L0 trigger system. The selection of the channels contributing to a particular Super Cell as well as the number of Cells is programmable: in order to match with the foreseen granularity of the LKr L0 trigger system (section 1.1.15.5), Super-Cells composed of either 4x4 or 2x8 cells have been considered. The readout of the sums data is performed via commercial serializers and standard Ethernet cable (cat.5) at 640 Mbit/s per link. The maximum number of links is 4, allowing up to 4 Super Cells from each CREAM module.

### 1.1.15.2 Data handling inside the FPGA

#### Zero suppression

The data copying mechanism described above will also allow to read interesting events without zero suppression, in order to have, at a later stage of the analysis, all the original data available. This scenario will allow for a non-zero suppressed readout while keeping a simpler network structure and a lower transfer rate. Reading only L1-selected data reduces the data rate by a factor at least 10, since the L1 rate is expected to be  $O(100 \text{ kHz})$ .

An additional data rate reduction is possible by applying a simple zero suppression algorithm to individual channels. Since for each event a large fraction of channels will only contain pedestal counts, one can discard channels where the difference between the maximum and the minimum value of the samples is below a predefined value (programmable and possibly different for each channel). This will add at least a factor 8 of data reduction, and it will reduce and simplify the configuration of the LKr event building farm.

Pre-processing at this stage could flag each sample, using the two free bits to pack a sample into a 16 bit data word, e.g. to mark whether it is consistent with a pedestal, or if it is the maximum sample, to speed up later processing. This could help in speeding up further processing, allowing to quickly search for interesting channels in the early phases of the processing, before any refined reconstruction. Header and packet formatting info should be added as well.

#### Possible additional computations

A more sophisticated use of the readout using the processing power of the FPGA could be investigated. Repetitive calculations on the data samples can be performed in the FPGAs rather than in the PC farm, thereby improving the throughput. A digital filter in the chain could perform computations on many samples and produce a result for each time slice. As an example, this type of filter could make raw computations of energy and time on each 25 ns sample, or it could also average the 8 samples and possibly flag the corresponding data as “pedestal”, using the 2 extra bits available in the DDR2 memory for each channel.

### 1.1.15.3 Network connections

#### Network architecture

We are working in the hypothesis that there is no need to transfer information at the single cell layer to the DAQ at the L0 rate. In the approach described above, selected L0 fragments are stored in an intermediate buffer, waiting for L1 to be transferred to the event building PCs. This hypothesis assumes also that the L1 rate is around 100 kHz. With this rate, no zero suppression and 10 s readout time, the aggregated throughput from each CREAM is 400 Mbit/s. Multiplexing the links from 16 CREAMs in local switches to one 10 GbE link does not limit the transfer rate: 28 x 10GbE links are used from the experimental hall to the NA62 control room. There a large switch ( $O(128)$  ports) houses the

28 links, a farm of multi-core PCs with 10GbE cards and a suitable number of 10GbE links for the transfer of complete events to the final NA62 event building farm.

The large switch in the NA62 control room is a part of the larger NA62 network infrastructure switch, both for cost reasons and for the capability of transferring LKr built events to the final event building farm using the switch backplane. The same switch could host the LKr trigger readout machines and additional service connections to the LKr readout control PC.

This implementation is dimensioned to allow a full-rate non-zero suppressed data transfer. The baseline mode of operation is however to apply a mild zero suppression to the events, except for a fraction of those (random events, calibrations, downscaled control triggers). In this scenario, one can gain on the number of event builder PCs. Special non-zero suppressed runs could still be performed at a reduced rate.

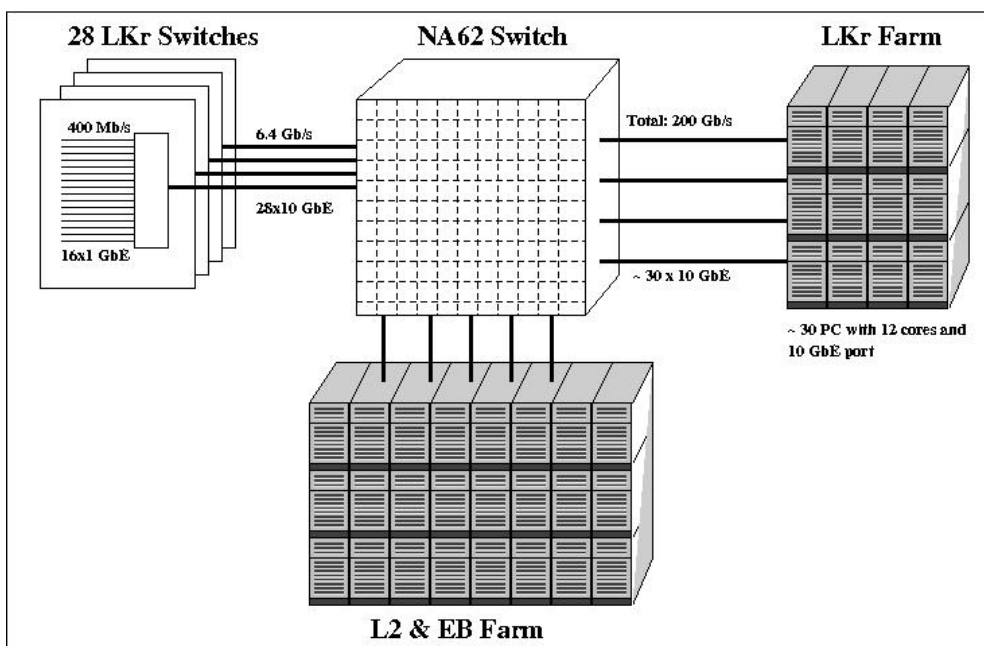


Figure 20: Layout of network connections and LKr PC farms.

## PC farm

The main function of the LKr PC farm (see Figure 20) is that of collecting the event fragments from the 432 CREAM boards and building one single LKr event out of them. This implies that for each L1 readout trigger (or a block of them), the LKr control PC will also distribute the address of the destination PC (as distributed from the L1 Trigger Processor). As additional tasks, the LKr PC farm could perform operations like “halo-expansion” of clusters (*i.e.* the flagging of cells which must be readout, independently from their sample content, because they are close to a region with cells above some threshold) and zero suppression (if not already done in the FPGA).

Assuming a figure of 4 ms to build a LKr event, about 30 PCs (12 cores each) are required to do all the LKr event building at an input rate of 100 KHz. Increasing this number will give more CPU power for the additional tasks described above or for pre-computation for the L2 trigger.

#### 1.1.15.4 Temporary Readout System

This system will be used until the end of 2012 for beam surveys and technical runs if needed.

The NA48 LKr readout system, based on CPD modules (17), was consolidated in 2008 to replace the optical links, the Data Concentrator and the VME RIO readout for maintenance reasons. Data from CPDs are read and stored in DDR2 memories in the Smart Link Modules (SLM) (18) and then read via Ethernet to a series of PCs. All activity on the LKr before the installation of the new readout described in this document relies on this CPD-SLM readout. The system was shown to be capable of reading non-zero suppressed data up to a trigger rate of 10 kHz. For operation of the LKr before the installation of the new NA62 system, the consolidation of the power supplies is needed, as well as the development of a trigger interface to connect the new NA62 trigger system with the old RIO-based LKr trigger system.

##### Existing hardware

The heart of the new system is the SLM. Each SLM receives data from the CPDs via a parallel 20 bit LVDS path implemented on multiple standard RJ45 cables. The readout protocol from the CPDs is handled by the SLM FPGA, and each event is stored in a 1 GB DDR2 memory. The interface with the outside world is via an Ethernet connection. The readout protocol is based on basic MAC packet transfer, and special protocol commands are implemented for the interaction with a PC. The typical sequence of network operations is the following:

- initialize the SLM at the start of spill;
- read the number of collected events at the end of spill;
- loop on the number of events, reading them one after the other and process them.

An internal timestamp is also implemented, with a counter incremented by a 1 MHz clock: this was intended to match fragments from different SLMs, but differences between quartz crystal frequencies in each SLM prevent the use of this feature.

It should be remarked that the NA48 LKr readout system requires the NA48 clock distribution system to be running, which should therefore be maintained for the interim period, and guaranteed to be phase-synchronous with the TTC one (*e.g.* by generating both from the same master source).

##### NA62 to NA48 trigger interface

In the SLM system the trigger distribution is essentially the one of NA48 (19). The trigger requests and information are received from the NA48 Trigger Supervisor, which will be not maintained for NA62. The 64-bit information arrives to the RIO-TIC processor through one differential PECL pair, and is managed by the TAXI chip (AMD Am7969). The NA62 LO Trigger Processor (replacing the NA48 Trigger Supervisor) communicates decisions and information through the TTC system. To allow interoperability, the trigger receiver stage of the old LKr readout system needs to be redesigned, in order to feed the CPD system with the TTC information. The easier way to do this is to use a TEL62 board, which is equipped with a TTC receiver (TTCrx), and build a new daughter-card for it housing a TAXI chip, implementing a protocol conversion (see Figure 21). The PP-FPGA and SL-FPGA in the TEL62 are used to re-encode the information, adding or modifying the data format to match the protocol specifications. In addition this new daughter-card contains other inputs, to manage XOFF signals (back-pressure control signals) from the read-out boards and to receive calibration triggers.

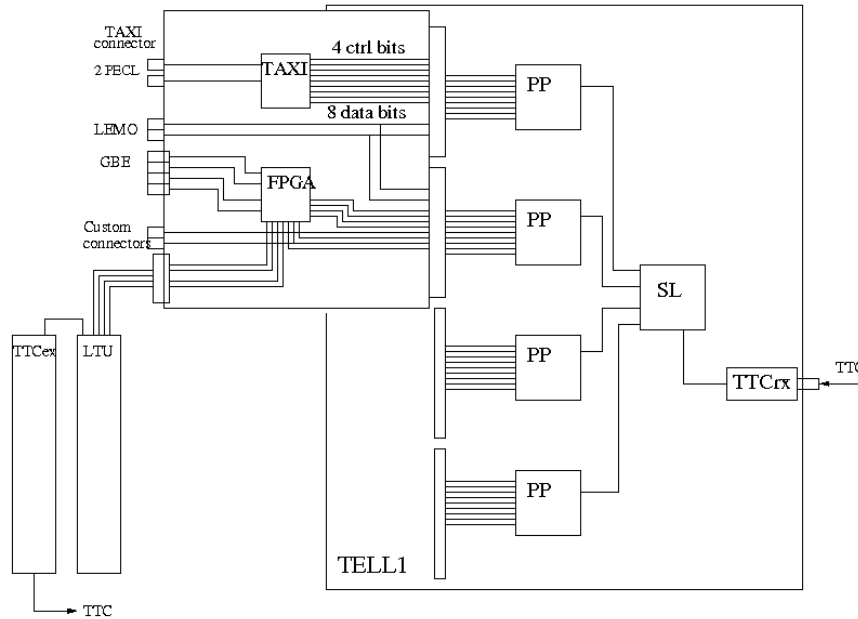


Figure 21: Scheme of the trigger interface for the SLMs.

Furthermore, in order to provide a small-scale standalone trigger supervisor replacement system, the new daughter-card hosts 4 Gigabit Ethernet cables to possibly collect trigger primitives from other detectors, and other inputs for direct connection of reference or trigger counters. The TEL62 FPGAs are used to implement the logic decisions. This functionality will be very useful to test the new trigger system and to provide a working system for test beams in the early phase of the data taking or during technical runs. The output from the card is in a format suitable for the LTU module, to provide the TTC trigger distribution to any NA62 sub-systems.

### Farming

The current LKr readout PC farm is composed of 12 Super Micro SC808T-980 PCs and of 2 (with a possible connection of two more) Elonex power servers. Each PC is equipped with one 4-port GbE interface card. The existing software is able to read at end of burst all the data from the SLMs, to re-format it, to optionally apply a zero suppression algorithm, to format the output data in a buffer and to write it on disk.

With this software the system was tested using two complete CPD racks, and it was shown that with the Super Micro machines (faster and with more memory) it is possible to read all events at a 10 kHz trigger rate. In addition, running on those machines multiple copies of the same program, it is possible to reduce the time needed to process a burst by a factor 2. With other PCs (slower and with less memory), the same results cannot be obtained: possible solutions are either their replacement or the installation of two additional PCs of the same kind, wiring their Ethernet cards to only 2 SLMs.

For the operation of a system in a possible NA62 run, several tools need to be prepared:

- a control mechanism to start centrally all the readout programs on all PCs;
- a data-merging mechanism to build complete events from the various fragments;
- a safe mechanism to match LKr events with event fragments from other detectors;

- a monitoring system to supervise and control the operations.

### CPD power supplies

The NA48 CPD power supplies were produced in the mid-80s with special low-noise switching supplies for the PS195 experiment and refurbished for the NA48 application, which required higher output currents. The actual power requirements are: +5.2V 350A, -5.2V 90A, -2V 8A, +15V 15A, -15V 1A. The power supplies have reached their expected life cycle and, in order to have an operational readout while the new CREAM system is being built, at least a fraction of them should be replaced with new ones, using the same mechanics and newer power modules. Requests have been sent and prototypes are going to be tested to verify the functionality and the coherent noise level.

### 1.1.15.5 LKr L0 Trigger System

#### Introduction and overview

The Level 0 LKr electromagnetic calorimeter trigger identifies electromagnetic clusters in the calorimeter and prepares a time-ordered list of reconstructed clusters together with the arrival time, position, and energy measurements of each cluster. As such, the system also provides a coarse-grained readout of the LKr that can be used in L1/L2 software trigger levels and off-line as a cross-check for the standard readout (see Figure 22).

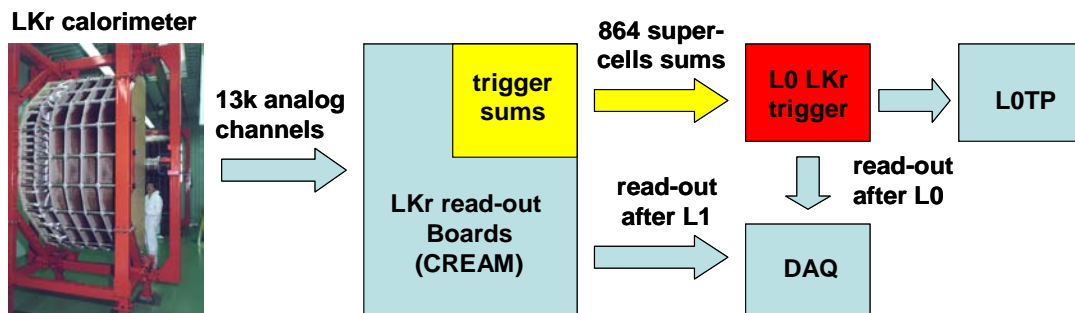


Figure 22: Scheme of the LKr L0 trigger system.

The trigger processor continuously receives from the LKr readout modules signals corresponding to tiles of 16 calorimeter cells (super-cells). Electromagnetic cluster search in the electromagnetic calorimeter is executed in two steps with two one-dimensional (1D) algorithms. From the trigger point of view the calorimeter is divided in slices parallel to the horizontal axis (assuming for the moment super-cells of 2x8 cells, 2 vertical, 8 horizontal). In the first step pulse peaks in space and time are searched independently in each slice with a 1D algorithm, along such axis. In the second step different peaks which are close in time and space are merged and assigned to the same electromagnetic cluster.

The LKr L0 trigger processor is a three-layer parallel system, composed of Front-End (FE) and Concentrator boards, both based on the TEL62 cards (Figure 23).

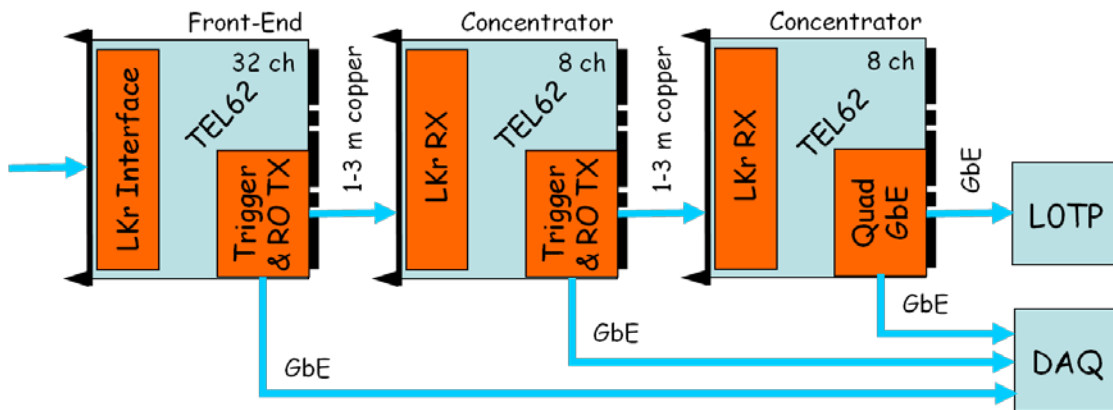


Figure 23: Preliminary implementation of LKr L0 trigger system.

Each FE board receives 32 tiles (Super Cells) from the LKr readout modules and performs peak search in space and computes time, position and energy for each detected peak<sup>15</sup>.

The concentrator board receives trigger data from up to 8 FE boards and combines peaks detected by different front-end boards into a single cluster.

The implementation of the trigger processor -assuming input tiles (Super Cells) of 2x8 calorimeter cells (2 cells along the horizontal axis and 8 cells along the vertical axis)- will be described here. The extension to the other configurations (8x2 or 4x4) is straightforward.

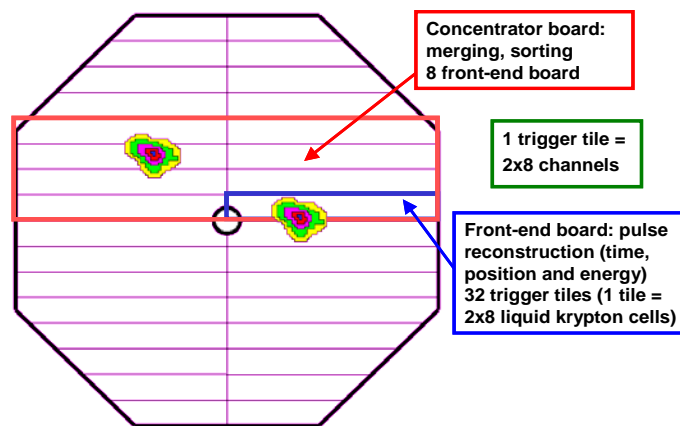


Figure 24: Trigger processor break-down

In total, the system will be composed of 36 TEL62 boards, up to 175 mezzanine cards and 192 high-performance FPGAs. A summary of the main parameters of the L0 LKr trigger is given in Table 14.

<sup>15</sup> In a previous concept the LKr readout system was assumed to provide analog sums, and ADC mezzanine boards were housed in the front-end boards.



Table 14: LKr L0 trigger parameters.

Input channels (tiles)	864
Trigger output channels	1
Readout output channels	28 (raw data) + 7 (reconstructed clusters)
Electronic modules (TEL62)	28 front-end+ 7 concentrator + 1 final concentrator
Latency	< 100 $\mu$ s

### Front-End boards

The Front-End boards continuously receive 864 trigger sums from the readout system, each one corresponding to 16 calorimeter cells. 28 FE boards are foreseen for the whole LKr L0 trigger system.

Each FE board receives 32 tiles trigger sums from a readout module, performs the peak search algorithm and transmits reconstructed peaks to the Concentrator boards.

Raw data received by the readout modules are also stored in L0 latency memories, to be readout after a positive L0 trigger is received.

The peak search algorithm is executed in parallel on all the tiles in the following steps (Figure 25):

- Peak search in space. A peak in space is defined by the following condition:  
 $E_{i-1}[n] < E_i[n]$  AND  $E_i[n] > E_{i+1}[n]$   
 where E is the ADC count, i is the tile number and n is the sample number.
- Peak search in time. A peak in time is defined by:  
 $E_i[n-2] < E_i[n-1] < E_i[n]$  AND  $E_i[n] > E_i[n+1]$
- Threshold check:  
 $E_i[n] > E_{th}$
- Parabolic interpolation in time around sample maximum, using samples n-1, n and n+1 to get an estimate of peak height ( $E_{max}$ ).
- Linear interpolation in energy between samples n-2 and n-1 to get the fine time corresponding to a programmable fraction between 0 and 1 of  $E_{max}$ .

Information on reconstructed peaks (maximum energy  $E_{max}$ , coarse time and fine time) are transferred from the Front-End board to the Concentrator boards on a low-latency dedicated trigger link.

A preliminary version of the above peak reconstruction algorithm was simulated and implemented on an ALTERA Stratix I FPGA (the device used as PP-FPGA on the original TELL1 boards, see section 1.1.5). With 12-bit pulse height resolution a non optimized version of the algorithm can process one peak (5 samples) at a rate in excess of 80 MHz, corresponding to 62.5 ns to process one peak. The maximum acceptable peak rate in a single PP-FPGA is thus 16 MHz for this model of FPGA<sup>16</sup>.

<sup>16</sup> The new TEL62 board will have more powerful devices.

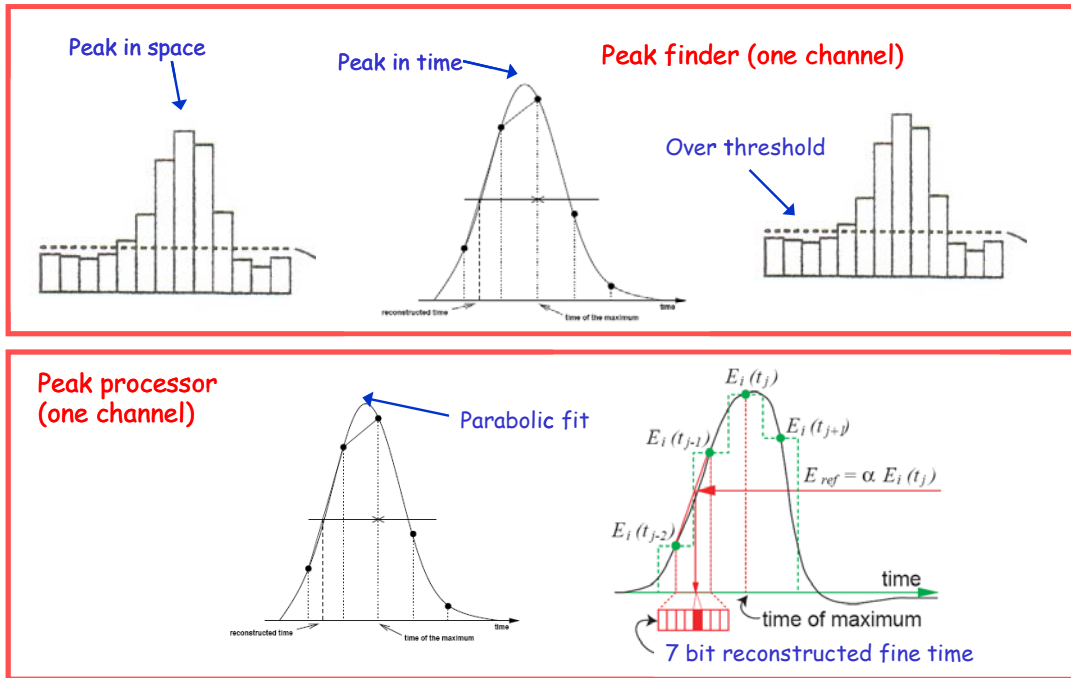


Figure 25 Peak reconstruction algorithm of LKr LO trigger.

Simulations were performed using the following pulse shape

$$A [ 1 + \sin(2\pi t/T - 3\pi/2) ]$$

with  $T = 175$  ns to check the algorithm, obtaining satisfactory theoretical performances. A more sophisticated algorithm will be implemented for the real data-taking.

Two custom mezzanines are foreseen for the FE TEL62 boards. The readout module interface mezzanine will continuously receive tile signals from the LKr readout modules. The transmitter mezzanine will transmit high-priority trigger data to the Concentrator boards on a custom trigger link, and low-priority readout data to PCs on a standard Gigabit Ethernet copper cable.

### Concentrator boards

The second-stage (Concentrator) boards receive trigger data from the FE boards, possibly combine peaks detected by different FE boards into a single cluster, and prepare time-ordered trigger primitives for the LO Trigger Processor.

Reconstructed clusters are also stored in LO latency memories and readout after a positive LO trigger signal is received via TTC. Eight concentrator boards are foreseen, the last one dedicated to the LO Trigger Processor interface.

Each concentrator board receives data from up to 8 FE boards, covering a region of 4 tiles along the vertical axis and 64 tiles along the horizontal axis. Each Concentrator board will:

- perform the peak reconstruction algorithm for clusters at the boundary between two FE boards around the central vertical axis of the calorimeter (a cluster in this region will be split between two neighbouring FE boards along the horizontal axis);
- merge information from different FE boards along the vertical axis, by associating reconstructed pulses along such axis to the same cluster. The region covered by a Concentrator board is divided in an inner region and an outer region: only clusters with a maximum along the vertical axis in the inner region are managed by the Concentrator board, to avoid double-counting of the same cluster (see Figure 335).

One custom mezzanine on each Concentrator board is foreseen to receive high-priority trigger data from the FE boards.

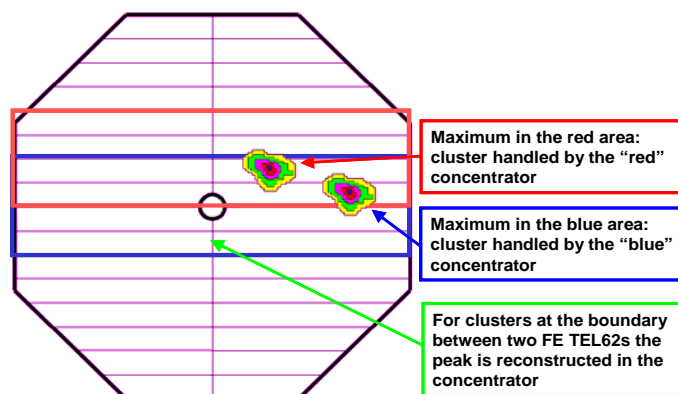


Figure 26: Concentrator board action.

### Connectivity and crates layout

Connectivity between LKr readout modules and FE boards is implemented with low-latency point-to-point links in such a way that a single FE board will receive a slice of 32 contiguous trigger sum tiles. 8 FE boards are connected to a Concentrator board, with some overlap between neighbouring Concentrators to guarantee that each cluster will be fully contained in at least one Concentrator board.

The LKr L0 trigger system will be hosted in one VME crate and two or three TEL62 crates. Two 21-slot TEL62 crates (like those built for the LHCb experiment) will host 14 FE boards (half of the calorimeter) each. The remaining eight Concentrator boards will be either divided between the two above crates or hosted in a third crate, depending on power consumption and cabling requirements.

### Readout modules interface

At least two tiles will be transmitted from each CREAM readout module, with 16 bits per tile transmitted every 25 ns to fully exploit the resolution of the CREAM on-board 14 bit ADCs (see section 1.1.15). Data can be transmitted from the CREAM to the FE boards using either optical fibres or copper cables: the decision will define the required mezzanine receiver card on the FE boards. A widely used solution has been identified, based on a Texas Instruments TLK2501 serializer chip and an optical transceiver. In this case 16 data bits will be transmitted together with two control bits. Assuming a 100 MHz transmission clock a bandwidth of 1.6 Gbps can be obtained, exceeding the minimum required bandwidth needed to transmit 16 bits at 80 MHz. A digital transmission option using copper cables is still under study.

## L0 Trigger Processor interface

Trigger primitives are produced by the Concentrator boards and sent to the L0 Trigger Processor. Each Concentrator board receives trigger data (portions of clusters in the calorimeter) from eight FE boards, but only clusters with a centre in the four inner FE boards are reconstructed by each Concentrator.

The reconstructed cluster rate for Concentrator boards covering the central region of the calorimeter can be (over)estimated as

$$\begin{aligned} \text{Reconstructed cluster rate} &= \\ &= \text{Instantaneous hit rate} \times \text{multiplicative factor for central regions} \times (4 / 28) = \\ &= 30 \text{ MHz} \times 3 \times (4 / 28) = 13 \text{ MHz} \end{aligned}$$

corresponding to

$$64 \times 13 \text{ Mbps} = 0.8 \text{ Gbps}$$

The average trigger primitive rate per Concentrator board can be estimated as

$$\begin{aligned} \text{Reconstructed cluster rate per Concentrator} &= \text{Instantaneous hit rate} \times (4 / 28) = \\ &= 30 \text{ MHz} \times (4 / 28) = 4 \text{ MHz} \end{aligned}$$

corresponding to

$$64 \times 4 \text{ Mbps} = 256 \text{ Mbps.}$$

Trigger primitives correspond to cluster multiplicities in a given time slot comparable to time resolution of the trigger processor. Time matching between different clusters (e.g. two simultaneous photons originating from the same particle) is done in the LKr L0 trigger system. A final Concentrator board collects and handles the trigger output of the seven LKr Concentrator boards. A single link connects the last Concentrator board to the L0 Trigger Processor.

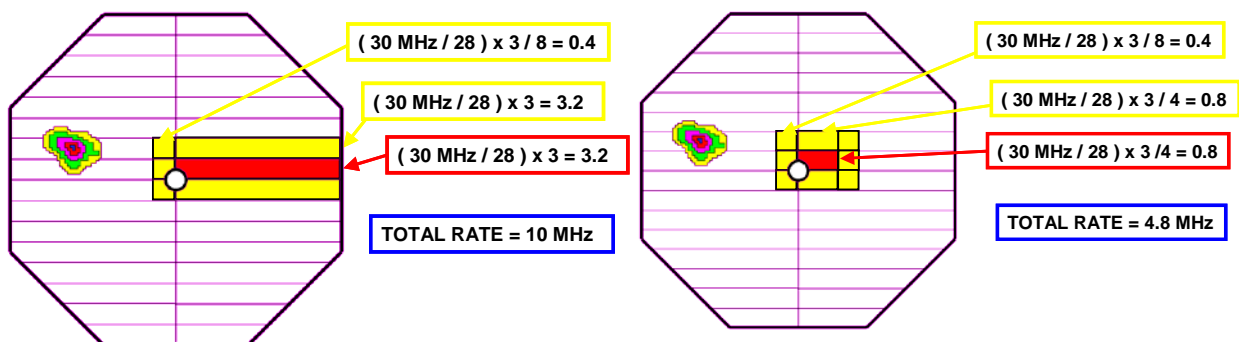


Figure 27 LKr L0 system hit rates

## Hit rates, dataflow and latency

The hit input rate is one of the most important constraints for the design of the LKr L0 trigger system, determining the minimum required computing power for each FPGA and the minimum required transmission link bandwidth for trigger and readout data. Hit rates were estimated under the following (conservative) assumptions: instantaneous design hit rate 30 MHz (1); all rates in the central region 3 times the average hit rate in the calorimeter, all particles (including muons) generating a shower of 256

calorimeter cells. Under these assumptions, and taking into account showers originating in the neighbouring regions, the rate for a FE board and for a PP-FPGA in a FE board are shown in Figure 27.

These rates must be matched to the PP-FPGA peak processing power and the various bus bandwidths inside the TEL62 board.

A new TEL62 output card was designed, housing two low-latency high-speed links for the trigger data, and two Gigabit Ethernet links for the readout data, in order to allow large (4.8 Gbps) data rates from the FE boards to the Concentrator boards.

The dedicated trigger link is based on a Camera Link cable assembly and a National Instruments 48-bit channel link chipset running at 100 MHz in one direction, and a 7-bit channel link chipset running at 66 MHz in the opposite direction. A high-quality industrial-grade halogen-free Camera Link cable is available, but relatively expensive. A custom cable solution with halogen-free individually-shielded twin-axial pairs assembled in the lab is being investigated. The bandwidth from a FE board to the Concentrator board is 48 bits at 100 MHz, while the bandwidth from a Concentrator to a FE board is 7/14 bits at 66 MHz.

Given the asynchronous architecture of the LKr L0 trigger system, latency is not constant and can thus only be estimated. Total latency can be divided in the following components: peak reconstruction in the FE PP-FPGA, data transmission from the FE PP-FPGA to the FE output connector, data transmission from the FE board to the Concentrator board over the dedicated trigger link, cluster reconstruction in the Concentrator board, data transmission from the Concentrator board to the L0 Trigger Processor. Each one of these tasks will contribute a few clock cycles to the total latency, giving a total latency of the order of few  $\mu$ s. Of course the total latency can increase in case of hit input rate fluctuations giving pile-up at the input of the processing elements and of the transmission links. A measurement of the capability of the system to absorb hit input rate fluctuations is thus given by the margin between the expected average processing and transmission capabilities at each stage of the processor and the maximum ones.

### **Readout**

After reception of a L0 trigger, the computed data are readout from the L0 LKr trigger to be eventually logged on tape together with sub-detector data. Two different kinds of data will be readout from the L0 LKr trigger:

- (a) raw data from each FE board and
- (b) reconstructed clusters (with reconstructed time, position, energy and shape) from each Concentrators board.

The readout bandwidth from each FE TEL62 can be estimated as:

$$\begin{aligned} \text{Readout BW (FE)} &= \text{L0 trigger rate} \times \text{tiles} \times \text{samples} \times 16 \text{ bit} = \\ &= 1 \text{ MHz} \times 32 \times 5 \times 16 = 2.56 \text{ Gbps} \end{aligned}$$

which can be easily matched to the maximum 2 Gbps maximum acceptable rate available from each FE board using some compression algorithm (to be developed).

The readout bandwidth from each concentrator TEL62 can be estimated as:

$$\begin{aligned} \text{Readout BW (Concentrator)} &= L0 \text{ trigger rate} \times \text{clusters} \times 256 \text{ bit} = \\ &= 1 \text{ MHz} \times 1 \times 256 = 256 \text{ Mbps} \end{aligned}$$

with very conservative assumptions of one reconstructed cluster per board per event, and a maximum of 256 bits to encode time, position and energy for a cluster in the calorimeter. This data can be used by the L1/L2 software trigger as a seed for more elaborate trigger algorithms.

### 1.1.16 SAC/IRC System

The design of the readout electronics is determined by the rate the detectors have to survive. The estimated rate for photons and muons in the SAC is about 1 MHz. With a signal length of 60 ns, the probability to have a second hit overlapping the first hit in the detector is about 6%. This is because the SAC, with its long attenuation length, can be considered as a single-channel detector. Considering the IRC with a rate of approximately 5 MHz the overlapping probability is about 25%. This means that the ability to distinguish two consecutive pulses is of great importance for the electronics design. A few possibilities can be envisioned: set a high threshold (due to the fact that most of the rate is caused by muons, one can set a threshold above the minimum ionizing particle (MIP) with a resulting loss of efficiency); construct a segmented detector (a reflective material, for example Tyvek, could be put between the individual segments in order to prevent crosstalk); or use a fast ADC readout (waveform digitizer) which will allow the observation of the different pulses by looking at the shape of the signal. The last solution has been chosen for both the SAC and IRC readouts.

The readout requirements for the SAC and the IRC can be met by means of high performance 1 GHz FADC operating as a mezzanine board on TEL62 general readout board designed for the LHCb experiment as shown schematically in Figure 28. One can effectively build such a system at a reasonable cost due to the availability of commercial 8 bit FADC chips operating at 1GHz.

At least two manufacturers have recently developed dual 8-bit 1 GHz FADC converters using CMOS technology<sup>17</sup>. These circuits can be operated as either as a dual channel 1 GHz or as one single 2 GHz FADC (2 GHz operation is possible using the leading and trailing edge of the 1 GHz clock. Every second input would not be used. The change from 1 to 2 GHz is made via the serial control interface). All features of the FADC are controlled with a serial interface. This eliminates many external components necessary for gain and offset tuning. For example it is possible to program the pedestals, gain and delay of each channel via this interface. The power consumption of both chips is in the order of 1.7 W and they are available in 20 x 20 mm LQFP packages. As presently the price of the circuit (4) is about a factor two lower and it has therefore been chosen for this application.

Figure 30 shows the schematic of four of the 16 channels of the GHz FADC mezzanine board. Only 3 ¼ integrated circuits are needed to implement 4 FADC channels. The mezzanine board will be equipped with 4 ADC and 2 FPGA circuits on each side of the PCB. The analogue inputs to FADC must be

<sup>17</sup> AT84D001B from [www.atmel.com](http://www.atmel.com) and FADC08D1000 from [www.national.com](http://www.national.com)

differential and the manufacturer recommends connecting the inputs with a 1:1 transformer to FADC inputs. The transformer can cause a shift in the pedestal of about 3 LSB levels at the signal rate of 3 MHz but this can be compensated with the FADC internal offset circuit and pre-samples of the pedestal. The transformer has a bandwidth of 0.4 MHz to 500 MHz. The input impedance is 100 ohms suitable for high quality twisted pair cables like CAT7<sup>18</sup>. The clock is distributed individually to each FADC with a low-jitter <1 ps rms clock fan-out circuit and must be AC coupled. The data outputs of each FADC are de-multiplexed to half to clock frequency (500 MHz) and are transmitted with LVDS levels to the frontend FPGA. One FPGA circuit can handle the outputs from 4 FADC channels. The task of this frontend FPGA is to reduce the rate of the FADC (8 Gbytes/s) to one suitable for transmitting to the TEL62. There are 160 data lines available in the 200 pin connector of the mezzanine board between the frontend FPGA and the PP-FPGA of the TEL62. A rate of up to 1.6 Gbytes/s can be transmitted to the TEL62 PP-FPGA per connector. The front-end FPGA has therefore to reduce amount of raw FADC data with a factor 5 by zero-suppression.

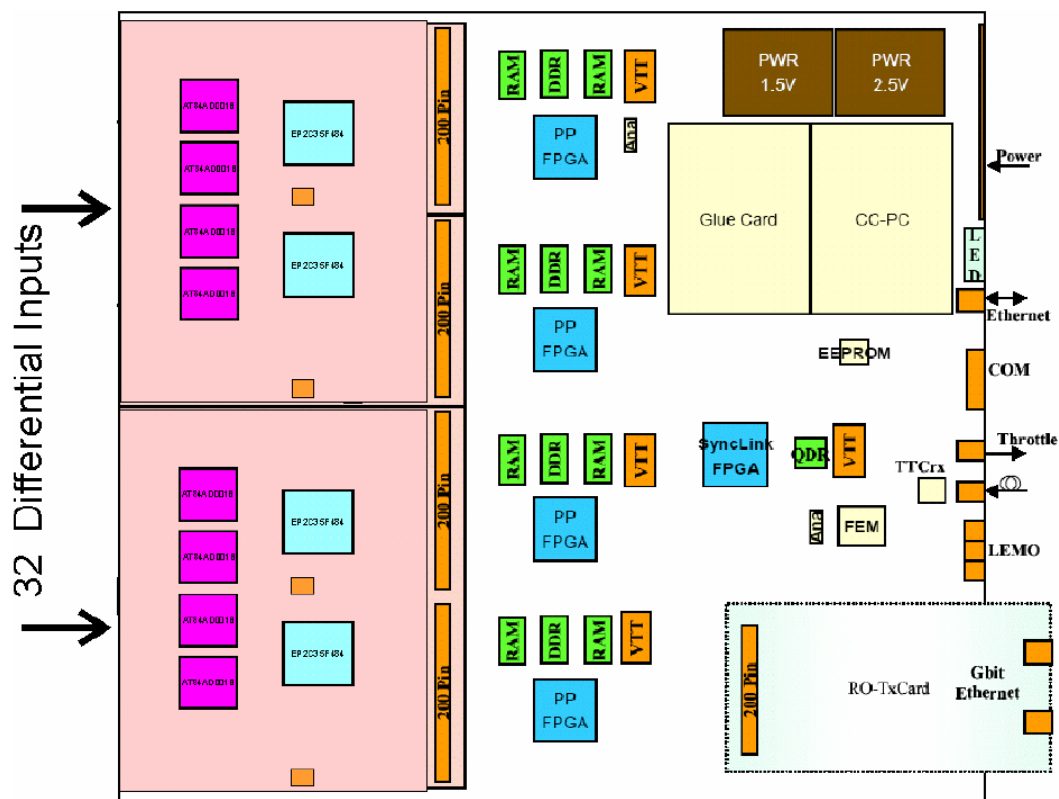


Figure 28 TEL62 board equipped with two FADC mezzanine cards (16 channels each)

### Front-End electronics

The amplitude of the signal from the PMT varies between 10 mV for minimal ionizing particle to approximately 3 V for a 75 GeV photon (a dynamic range of about 300). A frontend electronics is necessary in order to provide the necessary differential input to the FADC chip (see Figure 29). It will be

<sup>18</sup> For example, UNINET 7702 4P; [www.daetwyler.net](http://www.daetwyler.net) or [www.disdata.ch](http://www.disdata.ch) Art. Nr. 686350

placed on the PCB mounted on the PM itself. The requirements are the low power consumption since it will operate in vacuum (for the SAC) and the cooling might not be very effective. The signal will be transmitted to the FADC input with twisted pair cables.

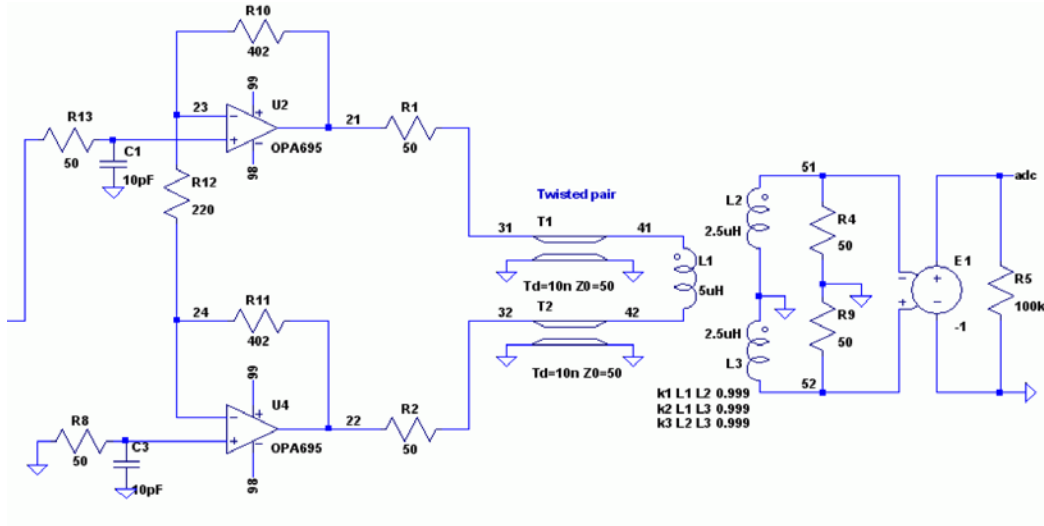


Figure 29 Frontend electronics and amplifier from the PMT to the ADC board.

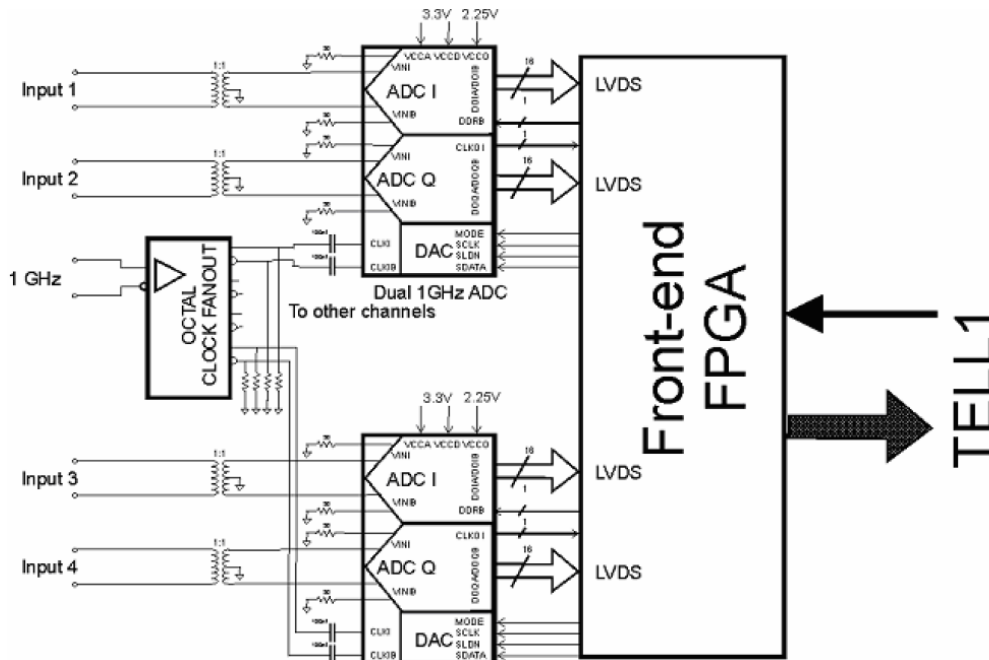


Figure 30 Schematics of the ADC board with four channels.



## Readout electronics

The SAC and the IRC will use a TEL62 based readout. The clock and the trigger will be distributed to the TEL62 using TTC. The data will be transmitted from the TEL62 to a readout PC and afterwards, using Gigabit Ethernet, to the central processing system for event construction.

### 1.1.17 GPU Improvements

A R&D program was started in Pisa to evaluate the possibility of improving the performance and cost-effectiveness of the TDAQ above the baseline solution described here, by massively exploiting GPUs (video card processors) in a hard real-time environment (20).

GPUs have the advantage over standard CPUs of a massive parallelism and huge computing power for parallelizable tasks which do not require complex control, and the advantage with respect to FPGAs of a much simpler programmability and scalability, being commodity devices. The present generation of devices can provide up to 1 Teraflop computing power with a memory bandwidth of 100 GB/s. Thanks to this performance the processing time heavily depends on the speed of data transfer towards the links to bring the data to the GPU.

While the use of GPUs in scientific computing is now quite established, its use in a data-acquisition system has not been tried, but preliminary studies within NA62 are encouraging in this respect and will continue.

A full discussion of this program goes beyond the scope of this document, and here we only mention some points where the use of GPUs could provide some advantage for NA62.

- GPUs could be hosted in L1/L2 PCs to provide a significant boost in the computing power for non-time-critical computations; this is straightforward.
- GPUs could be used to implement the L0 Trigger Processor, most likely requiring such device to be implemented in a PC, which is at present required to handle the GPU.
- GPUs could be used to replace the hardware systems evaluating L0 trigger primitives; this is the most challenging task, as the transfer and handling of all the sub-detector data is required, and a hard real-time response of both the GPU and the controlling CPU are mandatory. Some algorithms for computing L0 trigger primitives on GPUs have been developed and timed, showing that with due care to the hardware architecture involved the data processing capability and latency are not an issue.

## Bibliography

1. <http://ttc.web.cern.ch/TTC/intro.html>. *TTC: Timing, Trigger and Control Systems for LHC Detectors*. [Online]
2. **Sozzi, M.** A concept for the NA62 Trigger and Data Acquisition. *Internal Note NA62-07-03*. April 2007.
3. **Krivda, M.** Information on the ALICE trigger module. [Online] <http://epweb2.ph.bham.ac.uk/user/krivda/alice/>.
4. **Taylor, B.G.** *TTC laser transmitter (TTCex, TTCtx, TTCmx)*. User Manual (old TTCex version, new version in preparation, August 2010).
5. **Christiansen, J. et al.** Receiver ASIC for Timing, Trigger and Control distribution in LHC experiments. *IEEE Trans. Nucl. Sci.*, 43. 1996, pp. 1773-1777.
6. **Moreira, P.** QPLL manual; <http://proj-qpll.web.cern.ch/proj-qpll/images/qpllManual.pdf>. *CERN-EP/MIC*. 2005.
7. TTCrq mezzanine board. *CERN-EP/MIC*; <http://proj-qpll.web.cern.ch/proj-qpll/ttcrq.htm>.
8. **C. Gaspar et al.** *DIM: Distributed Information Management System*. CERN. Geneva : s.n. <http://dim.web.cern.ch/dim/>.
9. **Haefeli, G. et al.** TELL1, specification fo a common readout board for LHCb. *LHCb note 2003-007 - http://lphe.epfl.ch/tell1/*. 2005.
10. **Wiedner, Dirk.** Optical 12 input Receiver Card IF14-1 for the LHCb TELL1 Board. *CERN / EDMS*. [Online] 2006. <https://edms.cern.ch/document/758517/1>.
11. **Collazuol, G. et al.** *Proc. 11th EUROMICRO Conference on Digital System Design Architectures, Methods and Tools*. Parma 3-5 Sept.2008 : s.n., 2008 (DSD08). 405, IEEE.
12. **Christiansen, J.** HPTDC, High Performance Time to Digital Converter version 2.2. *CERN-EP/MI; March 2004*; <http://tdc.web.cern.ch/tdc/hptdc/hptdc.htm>. 2004.
13. **NA48 Collaboration; Anvar, S. et al.** The Beam and Detector for the NA48 neutral kaon CP violation experiment at CERN. *Nucl. Instrum. Methods A 574*. 2007, pp. 433-471.
14. **Angelucci, B.** *Master thesis, University of Pisa*. Pisa : s.n., 2010.
15. **Christiansen J. et al. (RD12 Collaboration).** TTCrx reference manual. *CERN EP/MIC note*; [http://ttc.web.cern.ch/TTC/TTCrx\\_manual3.9.pdf](http://ttc.web.cern.ch/TTC/TTCrx_manual3.9.pdf). 2004.
16. Analog Devices AD9252 data sheet. *Analog Devices*. [Online] 2010. [http://www.analog.com/static/imported-files/data\\_sheets/AD9252.pdf](http://www.analog.com/static/imported-files/data_sheets/AD9252.pdf).
17. **B. Hallgren et al.** For the CPD (Calorimeter Pipeline Digitizer module) see: The NA48 LKr calorimeter digitizer electronics chain - Wire Chamber Conference 1998, Vienna. *CERN preprint EP/98-48*. 1998.
18. *The NA62 Liquid Krypton Calorimeter Data Acquisition Upgrade*. **Hallgren, B., et al.** Dresden : IEEE, 2008. NA62 note NA62-08-04, presented at the IEEE, NSS 2008, Dresden.

19. **Z. Guzik et al.** RISC mezzanines for controlling data acquisition in the NA48 experiment at CERN. *Nucl. Instrum. Methods A452*. 2000, p. 289.

20. **G. Lamanna et al.** *GPUs for fast triggering and pattern matching at the CERN experiment NA62*. Vienna : s.n. Vienna Conference on Instrumentation 2010 submitted to Nucl. Instr. Meth. Phys. Res. A).



## NA62 Acronyms and Abbreviations

ADC	Analog to Digital Converter
AM	Absorber Module
APD	Avalanche PhotoDiode
ASIC	Application Specific Integrated Circuit
BCRST	Bunch Counter ReSeT
BEATCH	Program to provide coordinates of all beam elements as input for alignment
BEND	BENDING magnet or dipole
BIF	Barrier Improvement Factor
CALB	Sub-detector data format for calibration
CCPC	Credit-Card PC: commercial processor on TELL1/TEL62 boards
CEDAR	Cerenkov Differential counter with Achromatic Ring Focus: differential Cerenkov detector developed at CERN
CHANTI	Charged ANTI
CHOD	Charged HODoscope
CKM	Cabibbo–Kobayashi–Maskawa matrix
CM	Circulation Module
COLL	COLLimator
COND	Condition data of the detector (extracted from DCS)
CONF	configurations of the run, including active detectors, trigger configurations, beam conditions, etc.
COST	calibration parameters computed by calibration tasks running on (partially or fully) reconstructed data.
COTS	Commercial-Of-The-Shelf
CPD	Calorimeter Pipeline Digitizer module
CREAM	Calorimeter Readout Module
CTL	Chamber Trigger Logic for straws L0 trigger system
DAQ	Data Acquisition System
DCS	Detector Control System
DDR2	Double Data Rate SDRAM (memory chips)
DIM	Distributed Information Management system
DLL	Delay Locked Loop
DLS	Data Logging System
DM	Distribution Module
DPRAM	Dual Ported RAM
DRAM	Dynamic RAM
EB	Event-Building
ECN3	Experimental Cavern housing the NA62 experiment
ECRST	Event Counter ReSeT
EDX	Energy-Dispersive X-ray spectroscopy
EE	End of Ejection
EOB	End Of Burst
EOC	End Of Column option for GTK chip architecture

EoC	End-of-Column part in the P-TDC chip architecture of the GTK
EOF	End Of Frame
FADC	Flash Analog to Digital Converter
FE	Front-End
FEE	Front-End Electronics
FEM	Finite Element Model
FIFO	First In First Out buffer
FISC	Filament Scanner, a beam profile detector inside the beam vacuum system
FNAL-NICADD	Fermi Nat Lab - Photo injector Lab (18Mev electron linac)
FPGA	Field Programmable Gate Array
FR	Fast Reconstruction
GbE	Gigabit Ethernet
GIM	Glashow–Iliopoulos–Maiani mechanism which suppresses flavour-changing neutral currents)
GOL	Gigabit Optical Link transmitter
GPN	General Purpose Network
GPU	Graphic Processing Unit
GTK	GigaTracker
HALO	A beam simulation program to calculate muon HALO rates
HPTDC	High Performance Time to Digital Converter
HV	High Voltage
IRC	Intermediate Ring Calorimeter
JTAG	Joint Test Action Group protocol
L0	Level 0 Trigger
L0TP	Level 0 Trigger Processor
L1	Level 1 trigger
L1TP	Level 1 Trigger Processor
L2	Level 2 trigger
LAV	Large Angle Veto
LED	Light-Emitting Diode
LG	Lead Glass
LGTS	Lead Glass Test Station
LKr	Liquid Krypton calorimeter
LTU	Local Trigger Unit
LV	Low-Voltage
LVDS	Low-Voltage Differential Signalling
MBPL-TP	Dipole Bending Magnet with Tapered Pole
MEPs	Multi-Event Packets
MIP	Minimum ionising particle
MNP33	NA62 Experimental Magnet
MUV	Muon Veto System
NAHIF	North Area High Intensity Facility
NIM	Nuclear Instrumentation Module
NINO	Fast front-end preamplifier-discriminator chip developed by ALICE
NNLO	Next-to-next -to-leading order
PCB	Printed Circuit Board

PDE	Photon Detection Eff.
Pe	Photo-electron
PECL	Positive Emitter-Coupled Logic
PEI	PolyEtherImide
PET	PolyEthylene Terephthalate
PMT or PM	Photomultiplier Tube
POPOP	1,4-bis(5-phenyloxazol-2-yl) benzene organic scintillator
PP-FPGA	Pre-Processing FPGA in the TELL1/TEL62 boards
PPO	2,5-Diphenyloxazole organic scintillator
PTP	Para-TerPhenyl
PVSS	Object-oriented process visualization and control system by ETM (a commercial SCADA system)
QCD	Quanten Chromodynamik
QDR-II	Quad Data Rate II memory
QPLL	Quartz-crystal based Phase-Lock Loop
QUAD	QUADrupole
RECO	Data format for fully reconstructed events
RICH	Ring Imaging Cherenkov
SAC	Small Angle Calorimeter
SAV	Small Angle Veto
SDRAM	Synchronous Dynamic Random Access Memory
SEM	Scanning Electron Microscope
SiPM	Silicon PhotoMultiplier
SOB	Start of Burst
SL-FPGA	Sync-Link FPGA in the TELL1/TEL62 boards
SLM	Smart Link Modules
SM	Standard Model of particle physics
SPI	Serial Port Interface
SPR	Single Photoelectron Response
SRAM	Static Random Access Memory
SRB	Straw Readout Board
TCC8	Target Chamber Cavern upstream of ECN3
TDAQ	Trigger and Data Acquisition system
TDC	Time to Digital Converter
TDCB	Time to Digital Converter Board
TDCC-FPGA	TDC Controller FPGA in the TDC boards
TEL62	Trigger and Data Acquisition board developed for NA62, based on TELL1 design
TELL1	Trigger ELectronics for L1 trigger: readout board developed by LHCb
THIN	Data format for summary data from fully reconstructed events
TRIM	Steering Magnets for the Beam
TTC	Timing, Trigger and Control
TTCex	TTC encoder VME board developed by CERN
TTCrq	TTC receiver mezzanine card developed by CERN
TTCrx	TTC receiver ASIC developed by CERN
TURTLE	Trace Unlimited Rays Through Lumped Elements, a beam tracking and simulation program

VME	Electronic bus and rack standard
VTL	View Trigger Logic for straws LO trigger system
WE	Warning of Ejection
WLS	Wave-Length Shifting
WWE	Warning of Warning of Ejection