THE ALICE CENTRAL TRIGGER PROCESSOR

I.J. Bloodworth[1], G. Di Marzo Serugendo [2], D. Evans [1], G.T. Jones [1], P. Jovanovic [1],
A. Jusko [3], J.B. Kinson [1], A. Kirk [1], V. Lenti [4], M. Luptak [3], P.I. Norman [1],
L. Sandor [3], P. Vande Vyvre [2] and O. Villalobos Baillie [1], for the ALICE collaboration.

[1] Department of Physics and Astronomy, The University of Birmingham, Birmingham, UK
[2] CERN, European Organization for Nuclear Research, CH-1211 Geneva 23, Switzerland.
[3] IEP, Slovak Academy of Sciences and Safarik University, Kosice, Slovakia.
[4] Dipartimento di Fisica dell'Università and Sezione INFN, Bari, Italy.

Abstract

The principal features of the ALICE Central Trigger Processor are presented, as described in the draft User Requirement Document currently under discussion.

1. INTRODUCTION

The Alice Central Trigger Processor design must be completed and the electronics built within a short time from now. Many sub-detectors are now developing their front-end electronics, so it is necessary to define fully the interfaces between the Central Trigger Processor (CTP) and each of the sub-systems with which it interacts. This has been done following the guidelines of ISO 9001 as adapted by CERN. At present a User Requirement Document (URD) [1] incorporating earlier ideas has been written and is under review. Work is beginning on a Technical Specification, though this cannot proceed far until the URD has been frozen.

The ALICE experiment [2] aims to measure all the signatures currently being considered for the production of the Quark-Gluon Plasma. It will be the first time that all the signatures have been studied in a single experiment, and the requirement makes a number of important constraints on the system. In particular, the cross sections for different processes to be measured differ by several orders of magnitude, and therefore special action must be taken to allow simultaneous measurement of all of them.

The other principal challenge for the ALICE experiment comes from the very high multiplicities to be expected in Pb-Pb collisions at LHC energies. ALICE is designed to cope with a maximum rapidity density dN/dy of 8000, since there is a great uncertainty as to the expected rapidity density at LHC energies. This requirement has motivated the choice of a collection of detectors having very different properties as regards sensitive windows and readout times. These differ by more than two orders of magnitude between different sub-detectors. We have learnt that at RHIC the measured multiplicities have come out at the bottom end of the expected range (dN/dy of 555 ± 12 ± 35 at √s = 130 A GeV) [3], but this does not tell us very much about ALICE conditions, as a quite different part of the nucleon structure function is explored in the collision.

2. FUNCTIONS OF THE CENTRAL TRIGGER PROCESSOR

The principal functions of the CTP are summarized in the context diagram shown in figure 1.
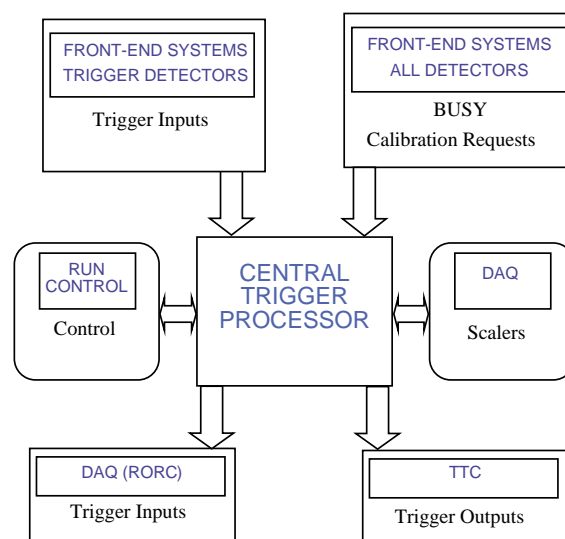
Fig. 1   ALICE CTP context diagram

The CTP is controlled by the ALICE run control. It receives trigger signals from a subset of (triggering) sub-detectors, and BUSY signals from all detectors. On the basis of this information, it sends to all the sub-detectors trigger signals which control the stages in the sub-detector readout. At the same time, monitoring information is collected and sent to the DAQ, both in the form of scaler information and in the form of more detailed information bundled into special "trigger event" records. In addition to the different types of physics triggers, the CTP also controls calibration triggers; these are submitted as requests to the CTP, which then sends the detector a

special calibration sequence, while protecting the detector from any competing physics triggers till the calibration sequence is completed.

Trigger information is checked and a decision made for each bunch crossing, i.e. at 25 ns intervals. In the case of ion running, where bunch crossings are separated by 125 ns [4], a mask can be applied to ensure that only filled LHC clock cycles are considered.

3. TRIGGER INPUTS AND LEVELS

The ALICE trigger system is currently organized into three trigger levels: **level 0** (L0), which acts after 1.2 µs and sends an early strobe to the front-end systems; **level 1** (L1) receives further information, and allows a re-classification of the trigger class at 5.5 µs; level **2** (L2) awaits the end of the longest detector sensitive period to determine whether the event should be rejected because of pile-up (see below for more details).

A third trigger level (L3), allowing partial reconstruction of events online in a PC farm, is being considered, but is beyond the scope of the present discussion.

The trigger inputs contributing to L0 and L1 are grouped according to their latencies. All and only signals which can be transmitted to the CTP electronics in under 900 ns from the time of the interaction are considered for L0. The L1 trigger receives all signals which are (*i*) too late for L0 and (*ii*) can be delivered in under ~5 µs, in order to deliver the signal to all sub-detectors in under 5.5 µs. The maximum numbers of inputs for each level, allowing for a possible expansion during the life-time of the experiment, have been fixed to the values shown in Table 1.

Table 1: Trigger inputs for each trigger level.

Level	Number of Trigger inputs
L0	16
L1	12
L2	4

The trigger inputs themselves are sent as signals in NRZ format (i.e. they retain their value throughout the full bunch crossing cycle.)

The CTP treats these inputs in three different ways. Inputs may be definitely required, explicitly not required, or ignored. The purpose of the "explicit no" is to veto

inputs rapidly. It is expected to be useful during setting-up of the experiment in order to check the complements of trigger conditions.

A sub-detector receiving a L0 trigger sets a BUSY immediately, and holds it until it is ready to read the next event. Note that in ALICE the sub-detector dead times are all independent.

4. TRIGGER CLASSES

The input and output sides of the ALICE trigger are logically linked through *trigger classes*. In order to define a trigger class, a collection of input and output requirements must be specified. For the input side, the status of the trigger inputs at each of the three trigger levels (according to the three-input-state description given above), must be specified. On the output side, the required sub-detector set for the class is to be specified. For a trigger to be issued, all the sub-detectors in its list must be ready, and all must be clear of pile-up. This is determined by specifying a past-future protection window for each sub-detector. The past-future protection window for a given class is defined to be the longest past-future protection for any detector in the class.

The protection interval specified for the trigger class then serves to determine the future of that class. If, after an event has taken place, a second event arrives within the protection interval for a class, the class is rendered invalid.

It is possible for an interaction to satisfy the input trigger conditions for more than one trigger class at the same time. In this case all the relevant trigger classes become active. Some may be eliminated subsequently if further interactions occur before the expiry of their protection intervals. The final (level 2) trigger is then sent to all the detectors listed in the surviving trigger classes.

A further refinement of this idea is required in the case of pp interactions. Here the lower multiplicities mean that some degree of pile-up can be tolerated in the detectors. In order to accommodate this, past-future protection circuits keep track of how many interactions occur within a given time-window centred on the triggered interaction, and flag a rejection if a pre-specified maximum is exceeded.

The use of trigger classes may be illustrated by the example shown in figures 2 and 3. For simplicity, only level 0 trigger inputs are considered. Figure 2 shows (top) a set of trigger inputs, which may be matched against the requirements for a number of trigger classes. If all the required inputs are found, and none is contra-indicated, the

class is accepted. On output, the sub-detector lists for a given class are matched against the BUSY/protection status of the detector. In the example given in figure 3, two detectors (the CPV and the TPC) are found to be BUSY. Comparison with the lists for three different trigger classes shows that only in one case (class 2) are all the required sub-detectors ready. In this example, trigger class 2 is selected on input, and all its sub-detector requirements are met, so a trigger is issued to each of the detectors in the class 2 list.
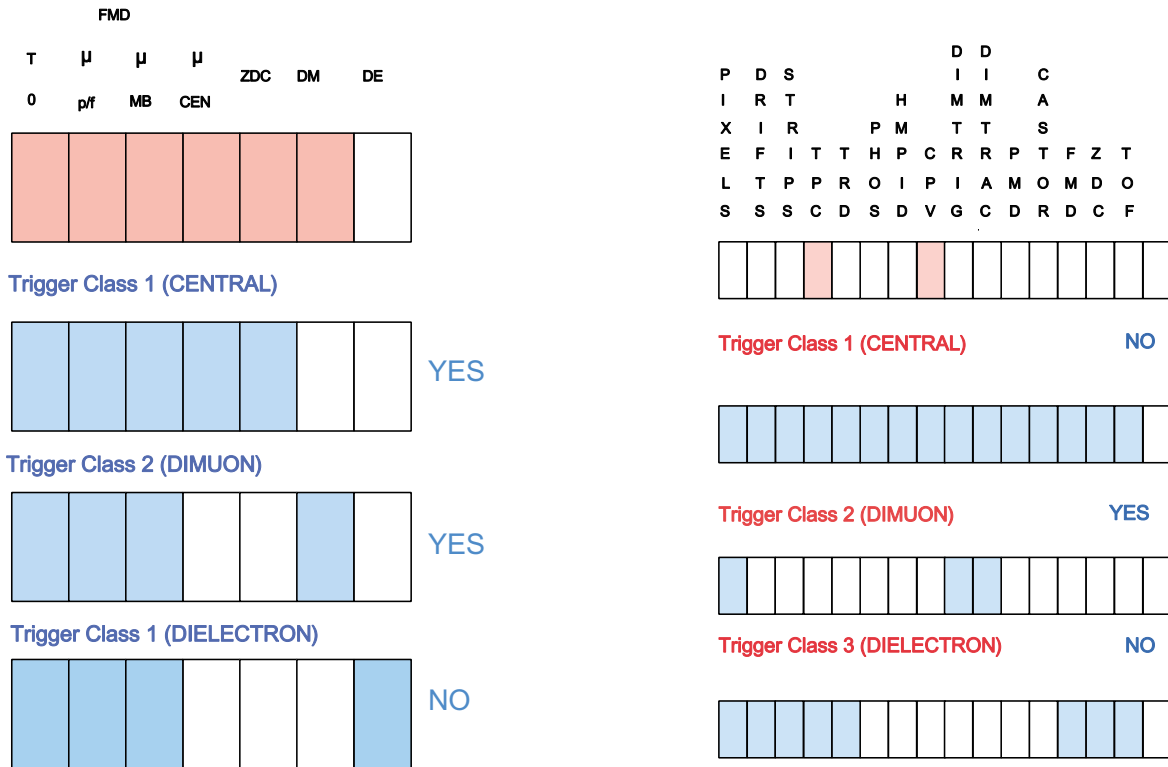

Fig. 2-3 Examples of trigger classes on input and output.

5. CALIBRATION TRIGGERS

All sub-detectors require calibration triggers, but these are relatively infrequent compared with physics triggers. The ALICE CTP always generates calibration triggers inside the "long gap" of 3.17 μs inside each LHC orbit. A sub-detector requests a calibration trigger via software. A program arbitrates in case of simultaneous requests and the trigger is performed during the first available calibration slot.

The sequence of signals sent to a sub-detector in the case of a calibration trigger is as follows: first, a pre-pulse is sent, which can be used by the sub-detector, for example, to activate its calibration circuit; a fixed time later, the usual L0-L1-L2 sequence is initiated.

All the calibration requests share a single reconfigurable trigger class, in which such parameters as the detector set to be read out and the past-future protection conditions to be applied can be specified for each new calibration. Note that the past-future protection conditions, if applied, mean that the success of a calibration trigger cannot be guaranteed. That is to say, as a result of the past-future protection conditions specified for a calibration trigger, the L0-L1-L2 sequence following the pre-pulse may be missing or truncated. The CTP does not report failed calibration triggers. Instead, at sub-detector level, two courses of action are possible. It may be sufficient to account for failed calibration triggers on a statistical basis simply by increasing the number of triggers required. Alternatively, if a strict count of calibration triggers is required, the condition can be readily monitored in the Local Trigger Crate (see below) since the pre-pulse which flags the calibration sequence is always delivered.

6. MONITORING

Monitoring of the CTP will proceed via two main methods: recording of scalers and the production of snapshots.

The normal operation of the trigger is performed by counting, using scalers, all the inputs and outputs of the CTP and by timing all the BUSY contributions. These are read out at regular intervals on a "round-robin" basis, and thus in general not simultaneously. However, in order to trap specific errors, it is possible to start and stop the scalers at fixed points, and thus force a correlated count.

A listing of bunch crossings containing interactions is issued for every orbit. This is done in order to allow the offline reconstruction software to compute the positions of "ghost" vertices from pile-up interactions in the case of pp interactions.

The other major monitoring tool is the "snapshot". For a period of a few milliseconds, the trigger data from each bunch crossing are recorded, whether a trigger was issued or not. This allows the trigger algorithms to be checked offline and gives an indication of the backgrounds behind accepted triggers.

The backgrounds to accepted triggers can also be monitored by the use of random triggers, which can be issued at any time and for which all inputs and outputs are recorded. A special trigger class is reserved for random triggers. In this case, detectors are also read out, giving further information as to how the backgrounds are caused.

# 7. TRIGGER OUTPUTS AND THE TTC SYSTEM

The basic output of the trigger system is the set of trigger pulses for levels 0, 1 and 2. Because of the critical latency for level 0, fast coaxial cables (4 ns m^{-1}) are used. The other levels are sent using the TTC system [5]. The level 1 trigger uses the TTC channel "A" [6]. As this level has a precise latency, the bunch crossing in which it arrives can be used to label the event. In ALICE each detector receives a different stream of triggers, depending on the classes for which it is included. This makes event labelling by orbit and bunch number an easier choice to implement than labelling by trigger number, as done in other experiments.

The level 2 triggers and the calibration trigger pre-pulse are carried by the TTC channel "B", as is the *orbit reset* signal. The orbit reset and calibration pre-pulse signals come at fixed points in the orbit cycle and they do not overlap. They use the highest priority TTC inputs B-Go<0> and B-Go<1>. The level 2 trigger must be sent with lower priority, and for this reason cannot be guaranteed to have a fixed latency. (Separate signals are sent for L2 accept and L2 reject, using B-Go<2> and B-Go<3>.) The level 2 trigger may jitter by up to a few microseconds if it collides with a higher priority signal. ALICE makes a fairly heavy use of the TTC facilities. The system is currently under evaluation.

In addition to these pulses, a certain amount of digital information is sent. In principle, the TTC system broadcasts an event number following every level 1 (channel "A") trigger; in the ALICE case this can be modified to be the orbit number, though in practice it is proposed that the number be counted locally at the TTCrx receiver by an associated FPGA, as this reduces the load on the "B" channel transmissions. In addition, eight further bits can be sent to define a trigger type. This is discussed in the following section.

8. LOCAL TRIGGER CRATE

The use of the "trigger type" bits is currently under discussion. They may be used to control the front-end operations, such as the suspension of zero suppression. In general, the actions for each sub-detector may be different, and for this reason, an interface is required to adapt the trigger commands appropriately. In principle, the maximum information available from the CTP at level 1 is the full set of active trigger classes. This information, together with any local information available to the sub-detector, can be used to determine which trigger type bits should be sent on the local TTC system. In order to do this, an additional board, Local Trigger Unit (LTU), can be placed in a VME crate assigned to each sub-detector, which translates the CTP information into the signals to be transmitted by the TTCvi. The transfer of signals between the CTP and the LTU is shown schematically in figure 4. Additional applications of the Local Trigger Crate, such as its use in monitoring the status of calibration triggers, are being considered.
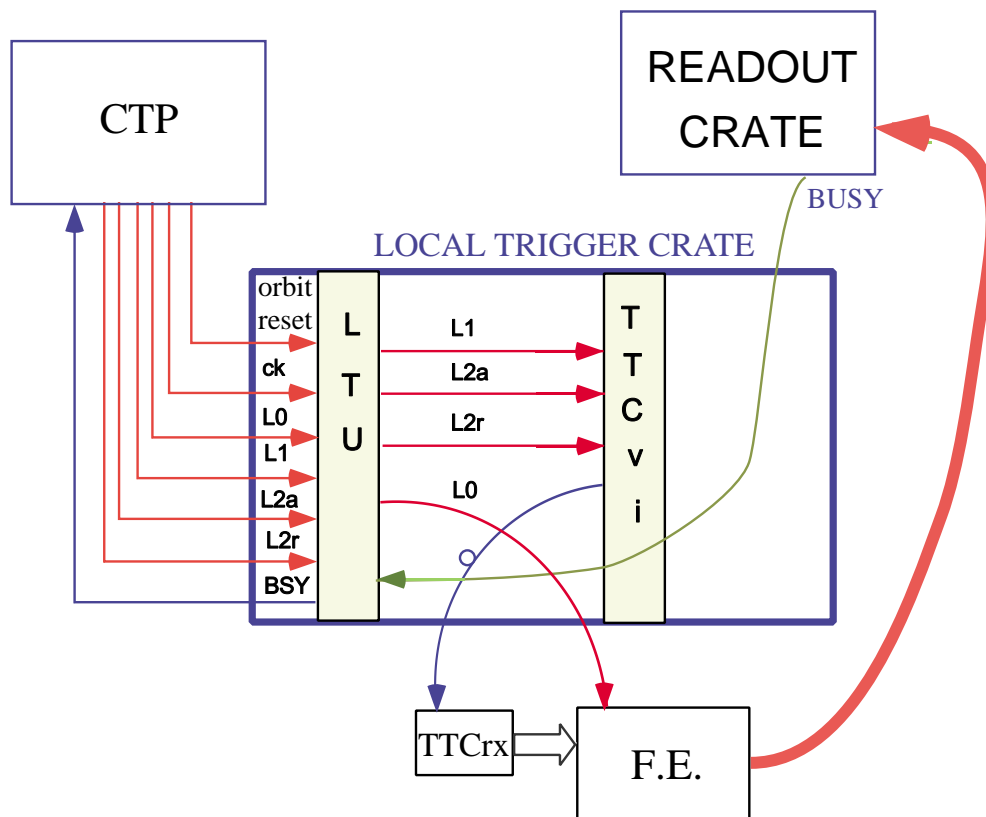
Fig. 4 Connections between CTP and Local Trigger Crate.

A useful by-product of the Local Trigger Unit is that it allows the sub-detector to be operated in stand-alone mode. Using simple logic, the LTU can be driven by a locally produced set of pulses, and these allow the sub-detector to operate independently. In this way, the sub-detector can be tested, using a standard environment, well before it is installed in the experimental area.

9. SUMMARY

The ALICE experiment uses a variety of detectors having significantly different sensitive windows and readout times. This leads to a trigger logic in which past-future protection is important, and where the proposed mode of operation involves independent dead times for each sub-detector. A draft User Requirement Document describing the Central Trigger Processor functions has been issued, and is under discussion. A Technical Specification is in preparation.

10. REFERENCES

[1] Central Trigger Processor draft User Requirement Document. http://www.ep.ph.bham.ac.uk/user/pedja/alice/urd01.ps

[2] ALICE Technical Proposal. CERN/LHCC 95-71 LHCC/P3

[3] B.B. Bock et al. hep-ex/007036; submitted to Physical Review Letters.

[4] LHC conceptual design. CERN/AC 95-05 (LHC)..

[5] B.G. Taylor IEEE Trans. Nucl. Sci. 45 (1998) 821
M. Ashton et al. CERN/LHCC 2000-002

[6] Ph. Farthouat and P. Gällnö, TTC-VMEbus Interface TTCvi – Mk II User Manual http://ttc.web.cern.ch/TTC/TTCviSpec.pdf