Track-Segment Sorting in the Trigger Server
of a Barrel Muon-Station in CMS

I. D’Antone, G.M. Dallavalle, S. Marcellini, A. Montanari, F. Odorici, G. Torromeo

I.N.F.N., Dipartimento di Fisica dell’Università, Bologna, Italy

Abstract

The Trigger Server of a muon station is part of the muon trigger chain of the DTBX chambers in the CMS barrel. The Trigger Server has to select in the $R\phi$ view the two best track-segments in the station, and transmit them to the Regional Trigger. The functional requirements on the selection mechanism are outlined, both for physics issues and technical aspects. Two fast sorting algorithms are being investigated with either sequential or parallel best and next-to-best search.
1 The DTBX trigger chain: a reminder

The basic components of the trigger electronics for the muon DTBX chambers (Drift Tubes with Bunch-Crossing (BX) identification) in the CMS barrel [1] have been described in details by M. De Giorgi et al. [2]. The performance of various parts of the DTBX trigger chain was studied by simulating muon tracks with a uniform spatial distribution and transverse momentum ranging from 4 to 1000 GeV/c. Effects due to geometrical inefficiencies, δ-rays, CMS magnetic field, drift velocity and electromagnetic showering were investigated. The trigger chain was simulated with both a functional model (using a FORTRAN program) and a behavioural model (using VERILOG), obtaining similar results [2].

The DTBX local-trigger chain consists of three parts: the Bunch and Track Identifier (BTI), the Track Correlator (TC) and the Trigger Server (TS). The Trigger Server is a system of decisional logic, used to select tracks in both the θ and ϕ view of a barrel muon station. While the TS for the θ views simply consists of a wired-or coincidence logic, the TS for the ϕ view is much more complicate. In the following discussion, only the Trigger Server in the ϕ view will be considered.

1.1 The BTI

The BTIs are trigger front-end devices [2, 3] which in the ϕ view provide triggering track-segments using a mean-timer algorithm independently in the outer and inner super-layer (SL) of a muon station (Fig. 1). Each SL is a stack of 4 planes of drift tubes parallel to the beam axis. Neighbouring planes are staggered by a half tube. Each BTI receives inputs from 9 tubes (3+2+2+2 in the 4 planes of a SL). The BTI generates a trigger when a track candidate has at least three hits - aligned along a valid track pattern within a programmable angular acceptance - over the 4 planes of drift tubes connected to that BTI. Trigger tracks with 3 aligned hits in a BTI are low quality triggers (L triggers); trigger tracks with 4 aligned hits in a BTI are high quality triggers (H triggers).

With a clean muon track the alignment condition is verified after a fixed amount of BXs thus enabling BX identification. From simulation studies the probability for a muon (with transverse momentum ranging from 10 to 1000 GeV/c) to produce a H trigger at the right BX is about 80%, a L trigger at the right BX is about 15%, a H trigger at a BX earlier than expected is about 2%. The off-time H triggers are generated by electromagnetic showering and delta-rays. For each H trigger found at the right BX, at least one off-time L trigger is found because of “ghost-hits” alignment which accompany a real track in a SL due to the left/right ambiguity of the drift tubes.

1.2 The TC

A device called muon Track Correlator correlates the tracks between inner and outer SL of the same muon station. The correlation test is very effective in rejecting ghost tracks. With the design proposed in [2], each TC has inputs from 20 BTIs: 5 BTIs of the inner SL and 15 BTIs of the outer SL (Fig. 2). The TC inputs overlap in the outer SL, where each BTI has three
output ports and is connected to three TCs. The TC accepts only tracks pointing to the inner BTIs. Notice that: i) each BTI included in a TC can give at most one track candidate per BX, with priority given to H triggers; ii) each TC of a muon station is assigned a radial direction, defined by an angle $\psi_R$ with respect to the normal to the station (for simplicity assume $\psi_R = 0$, i.e. the TC radial direction is normal to the station).

The TC algorithm described in [2] is shortly summarized in the following (see also Tab. 1). A BTI track has two coordinates: the angle $\psi$ with respect to the normal to the SL and the position $x$ along the SL central plane. On a BTI trigger every TC reads from the BTIs the coordinates of the triggering tracks.

In the first BX the TC looks for, separately in the outer SL and in the inner SL, the triggering BTI which contains the track (with priority given to H triggers) with $\psi$ angle closest to $\psi_R$ (min $\Delta \psi_R$). If there is an inner track and an outer track of equal quality, the inner track is taken. The “best track” thus selected is used to give a first measurement of the track parameters.

During the second BX the TC checks whether the best track in the inner SL is correlated with the best track in the outer SL. Correlation is found when the two tracks are aligned, i.e. when the $\psi_O$ angle of the outer track and the $\psi_I$ angle of the inner track are both equal to, within a programmable tolerance, the $\psi$ angle of the direction connecting the outer BTI to the inner BTI with respect to the normal to the muon chamber. If correlation is found the track parameters are recalculated using both tracks, in the third BX.

In pipeline to a first track, the TC also calculates, when possible, a second track. This second track selection is based on the min $\Delta \psi_R$, among all remaining BTIs in each SL, independently of the H/L trigger quality. The second track selection is not considered when a condition of “overlap trigger” occurs. This condition is generated by a BTI when two triggers are found in two consecutive BXs. A rough estimate of the overlap trigger rate, at the nominal LHC luminosity, corresponds to about one overlap trigger every 40 triggers [4].

1.3 The TS

The next step in the DTBX trigger chain is the Trigger Server of a muon station (Fig. 3). In the $R\phi$ view the TS has to select the two best tracks among the ones selected by all TCs of the station. These two tracks are transmitted to the Regional Trigger.

The number of TCs for a station can be quite large: currently 28 TCs are foreseen for the outermost muon station [4]. Each TC transmits to the TS its two best tracks serially, in two consecutive BXs (see Tab. 1). In fact, following [2] each TC sends in advance to the TS the $\Delta \psi_R$ value of the track being checked for correlation. The TS has to select (in pipeline) the two tracks in the station which are closest to the radial direction (min $\Delta \psi_R$), which should correspond to the higher transverse momentum tracks. The TS should then serially read the parameters (angle, position and quality bits) of the two tracks from the corresponding TCs, and send the two tracks to the Regional Trigger in parallel.

As shown in Tab. 1, both TC and TS need 5 BXs in order to complete their pipeline processing flow. However, thanks to the information handshake between TC and TS, the total latency added by the TC–TS processing chain to the DTBX trigger is only 7 BXs (Tab. 1). In the following we investigate more in detail how the track selection in the TS could be performed, taking into account the tight timing constraints.
2 Functional requirements of the TS Sorter

Every TC in a muon station can send to the TS up to two track-data ordered in quality [2]: the first transmitted track is the track with min $\Delta \psi_R$, with priority to H triggers; the second transmitted track is the track with min $\Delta \psi_R$ over all remaining triggers.

Let’s now define BUNCH1 and BUNCH2 respectively as the first and the second bunch of tracks arriving from the TCs connected to the TS (see Fig. 3) at two consecutive BXs. The TS sorting algorithm could be simple if it just looked for the best track of BUNCH1 and the best of BUNCH2, independently for the two bunches. However, it is not assured that the best track of BUNCH2 represents the second-best track among all BUNCH1 and BUNCH2 tracks. On the other hand, finding the true second-best track is particularly important in multi-muon events, as clarified in the following 2 examples:

i) in Higgs decays $H \rightarrow ZZ^* \rightarrow 4\mu$, assuming for instance $m_H \approx 130$ GeV, a muon from $Z$ decay and a muon from $Z^*$ decay have a large (16.7%) probability of entering the same $\Delta \phi$ sector of the CMS barrel muon-detector. Let us assume that the two muons produce two H triggers, one per TC in two far apart TCs, in each of the four muon-stations of the $\Delta \phi$ sector. In each station both TCs forward a good track to the TS in BUNCH1. Out of these two tracks the Trigger Server only selects one unless it also makes a next-to-best search within the same bunch. Over the 4 muon-stations of the $\Delta \phi$ sector it can happen that only two out of four good track-segments per muon are retained, thus making possible for the track finder of the Regional Trigger to discard both muons. Besides, the remaining two muon tracks experience a similar bias in the opposite $\Delta \phi$ sector, and the event can fail triggering;

ii) in Higgs decays $H \rightarrow ZZ \rightarrow 2\mu2\nu$, when $m_H > 500$ GeV, because of the large boost, the two muons from $Z$ decay have a sizeable probability of entering the same $\Delta \phi$ sector of the CMS barrel muon-detector. As in the above example it can happen that over 4 muon-stations only two track-segments per muon are retained, thus again making possible for the Regional Trigger to discard both muons. Electromagnetic showering, $\delta$-rays and acceptance holes decrease the trigger efficiency on such events even further.

We conclude that the Trigger Server should preferentially sort the true second-best track among all BUNCH1 and BUNCH2 tracks. In order to achieve this, the Trigger Server selects, among the tracks of BUNCH1, the first-best track (FBT) and the next-to-best track. On the following BX, the second-best track (SBT) is selected between the best of BUNCH2 and the next-to-best of BUNCH1. Therefore the sorting algorithm is applied in pipeline at each bunch. For normal events, the TS is able to sort FBT and SBT within two BXs. In case of overlap triggers, the TS is able to provide to the Regional Trigger at least the FBT data resulting from the sorting of BUNCH1.

Another important aspect to investigate is which informations the Trigger Server needs from the Track Correlators. The $\Delta \psi_R$ values from the two bunches of tracks are not sufficient. In fact, if all BUNCH1 tracks from the TCs were H triggers, “best track” would essentially mean the track closer to the radial direction (min $\Delta \psi_R$). However not necessarily all BUNCH1 tracks are H triggers, and since complete consistency with the TC algorithm is required, H triggers in
BUNCH1 must have priority. This implies that the TS has to know from each TC whether it is dealing with a H or L trigger. An extra bit of information should be provided by each TC together with the $\Delta \psi_R$ data.

Let’s now consider the particular sorting problem caused by off-time triggers, i.e. BTI triggers not associated to the right BX. In particular, from simulation studies [2], off-time H triggers predominantly occur one BX earlier than the correct one. Thus in the track transmission sequence from the TCs to the TS it can happen that, because of BTI triggers in two consecutive BXs, one (or more) TC is off-time with respect to the majority of the TCs, and/or there is a condition of overlap trigger in one (or more) TC (see Tab. 2). Flexibility in dealing with these cases implies that the TS sorter has to know for each TC whether it is dealing with a BUNCH1 or a BUNCH2 track. This extra bit of information should be provided by the TC together with the $\Delta \psi_R$ data.

Regarding H triggers, there is a further potential problem when a TC generates a “track reflection” in a nearby TC. Suppose a muon has generated a H trigger in a BTI of the inner SL and a H trigger in a BTI of the outer SL. Whilst the inner track is uniquely assigned to a correlator (say TCn), the outer track could be assigned to both TCn and TCn+1 (or TCn-1) (see Fig. 2). The magnitude of the effect depends on the angular tolerance assigned to the outer BTI track. This mechanism produces in BUNCH1 two tracks for the same muon which are passed to the Trigger Server. Of course the possibility of selecting both of them as best and next-to-best is not satisfactory. In first approximation next-to-best sorting should exclude TCs close to the TC containing the best track. In a neater approach, the TS sorter has to know from the TC whether a track belongs to the inner SL or to the outer SL, or whether there is a correlated track in the other SL. The correlation information can be provided by the TCs and, at the expenses of increasing the total TC–TS latency by one BX (see Tab. 1), it could be used in the track sorting process by assigning higher priority to the correlated tracks. However, in general the correlation information is only useful when at least one of the two tracks is a H trigger. In fact, as explained in [2], the probability of a BTI giving a L trigger at the right BX in one SL is only 15%, so that the probability of two correlated L-trigger tracks is quite low, at most 2%. Even when one of the two tracks is a H trigger, the usefulness of the correlation information is further diluted because of the geometrical acceptance. Simulation studies [2] show that the probability of having two correlated H triggers along a muon track in one TC is only about 40%, mainly because of a poor geometrical acceptance. Therefore the most relevant information for the TS sorter seems to be whether a given H trigger belongs to the inner SL or to the outer SL. Further simulation studies are needed to better clarify this point. If necessary, this extra bit of information can be provided by the TC together with the $\Delta \psi_R$ data.

In summary, the basic functionality for the TS of a muon station is to sort the two best tracks (FBT and SBT) in the following way:

- sort the best track in BUNCH1, in order to determine the FBT;
  sort (and store) the next-to-best in BUNCH1;
- search for the best track in BUNCH2 and compare it with the BUNCH1 next-to-best, in order to determine the SBT.
Furthermore, three quality bits should accompany the $\Delta \psi_R$ value of a TC track: a H/L trigger bit, a first/second bunch bit, a inner/outer SL bit (or a correlation bit).

### 3 Sequential or parallel sorting of best and next-to-best

In the following we study how a sorting procedure, fulfilling the functional requirements outlined in the previous section, could be implemented using the current VLSI technology (scale 0.7 $\mu$m).

The number of TCs in a station depends on the station under study. Outer stations have a larger number of drift tubes and therefore need more TCs than inner stations. Let us assume that the maximum number of TCs is 28. The station TS should then handle up to 56 tracks (28 tracks per BX in two consecutive BXs) and select the best two tracks for transmission to the Regional Trigger. A simple masking logic based on the quality bits previously defined can be used to give priority to classes of tracks. For the tracks surviving the masking, “best track” essentially means the track with minimum $\Delta \psi_R$, i.e. closer to the radial direction. Then, the TS sorter performs the following processing cycles:

- **Cycle1**: analyse the $\Delta \psi_R$ values of the 28 tracks of BUNCH1, search for the min $\Delta \psi_R$ (the FBT), search for and store the next-to-min $\Delta \psi_R$;

- **Cycle2**: analyse the $\Delta \psi_R$ values of the 28 tracks of BUNCH2, search for the min $\Delta \psi_R$ and compare it to the next-to-min $\Delta \psi_R$ of the previous bunch thus determining the SBT.

Each cycle must be completed within one BX, i.e. 25 nsec.

We have investigated two approaches to tackle the problem of finding the min and next-to-min values out of 28 data words:

- **SEQUENTIAL**: a fast min-search algorithm is applied twice over 28 data words;

- **PARALLEL**: a parallel min and next-to-min search algorithm is applied twice, in cascade over small groups of data words.

The response time of these algorithms has to be independent as much as possible from our hypothesis of 28 input words.

The sequential approach assumes that the TS min-$\Delta \psi_R$ selection is very fast, so that it is possible to mask the minimum and run the selection again for the next-to-min. Assuming that the $\Delta \psi_R$ values are 5-bit words, a bit-serial type algorithm for min-search is fast because the number of input words is larger than the number of bits per word. An algorithm of this kind is explained by the flow diagram of Fig. 4. In this algorithm the 28 TC data words are analysed in parallel one bit after the other, and the TCs are progressively deselected, finally leaving selected the TCs with minimum $\Delta \psi_R$. In the circuit implementation of the above algorithm we use 20 levels of logic gates, four levels per bit. Although this algorithm is fast for min search, it has to run twice in order to perform the next-to-min search during Cycle1. This is
The SBT search during Cycle2 is performed by injecting the next-to-min of BUNCH1 together with the 28 tracks of BUNCH2 in Cycle2, and by searching for a min $\Delta \psi_R$ over 29 tracks. Because of the next-to-min search, Cycle1 takes a longer processing time than Cycle2. From a VERILOG simulation, with synthesis for an ASIC chip, the estimated response time for the Cycle1 algorithm is 20 nsec.

In the parallel approach the sorting problem is tackled by dividing the input data words in groups and performing the analysis in a tree structure with two layers of processing units, as sketched in Fig. 5. The optimal division of the input words is found by minimizing the number of two-word comparators used for the min search. The 28 input data words are split into 7 groups of 4 words each. On each group, a Trigger Server Processor (TSP) unit is applied for searching in parallel for min and next-to-min. The functionality of the tree-structured processing circuit is:

- on Cycle1, each TSP4 in the first layer sorts the four input words from the TCs, transmits the min to the second TSPs layer, while the next-to-min is stored locally and carried over to Cycle2. The TSP7 unit in the second layer sorts the seven input words from the TSP4s, searching for min and next-to-min. The TSP7 min in this cycle is the FBT. The TSP7 next-to-min is stored locally and carried over to Cycle2;

- on Cycle2, each TSP4 in the first layer sorts the four input words from the TCs together with the Cycle1 next-to-min locally stored. Each TSP4 transmits the min to the second layer. The TSP7 unit in the second layer sorts the seven input words from the TSP4s together with the TSP7 next-to-min of Cycle1. The resulting TSP7 min in Cycle2 is the SBT.

The total number of levels of logic gates that we use in the above parallel sorting approach is 21. A response time of 11 nsec is estimated with the VERILOG simulation for an ASIC chip.

We conclude that, on the basis of the response time, the parallel approach should be preferred. This algorithm leaves more time available for the extra logic necessary for I/O buffering, synchronization and control. In the following the design of the TSP unit is described in detail.

4 TSP units and tree-structured processing

Each TSP4 unit (see Fig. 6) receives input words from 4 TCs. In the design of the TSP4 unit we assume that a TC word consists of 8 bits: a 5-bit $\Delta \psi_R$ word, a inner/outer SI bit, a H/L trigger bit and a first/second bunch bit. The functionality of the TSP4 unit develops in two consecutive BXs. On Cycle1, BUNCH1 data arrive from 4 TCs. The 4 $\Delta \psi_R$ words are compared and min and next-to-min are searched for in parallel. A “sel” line is activated, corresponding to the TC with the minimum $\Delta \psi_R$. The min-$\Delta \psi_R$ word is fed on output for processing in the next TSP layer. The next-to-min-$\Delta \psi_R$ word (TC carry) is stored for Cycle2. On Cycle2, BUNCH2 data arrive from 4 TCs. They are compared among themselves and with the next-to-min of Cycle1. If the min found in Cycle2 is just the next-to-min of Cycle1, a
"postsel" line is activated, corresponding to the concerned TC, otherwise a sel line is activated. The min-$\Delta \psi_R$ word in Cycle2 is fed on output for processing in the next TSP layer.

The comparison stage relies on a fast 2-word (5-bit each) comparator where 2-by-2 bits are compared in parallel, using 7 levels of logic gates. The component used is the standard "COMPM8 magnitude comparator", modified for 5-bit words. In the TSP4 unit 10 of such 2-word comparators are used in parallel: 6 comparators are used for the min search and 4 comparators are necessary for next-to-min search. A simple coincidence logic uses the results of the 10 comparators to find min and next-to-min in parallel. This logic also contains a priority rule to disentangle cases of equal data from 2 or more TCs, using 3 levels of logic gates. In total the min and next-to-min search in one TSP4 unit is implemented with 10 levels of logic gates. For an ASIC chip, according to the VERILOG simulation, the response time of this circuit is about 5 nsec. The TSP7 unit of the second layer uses 28 comparators in parallel. The coincidence logic of min and next-to-min in TSP7 uses 4 levels of logic gates. The total number of levels of logic gates used in the parallel sorting approach is therefore 21.

Additional logic is used to control the I/O between TSPs and TCs units:

- the first/second bunch bit in each TC data word is used to decide whether the TSP should perform Cycle1 or Cycle2 actions. An example of a track transmission sequence from TCs to TS with the corresponding sorting sequence in the TS is shown in Tab. 2;

- situations like off-time and overlap triggers are dealt with through a masking logic as shown in Tab. 2: when a mixture of BUNCH1 and BUNCH2 tracks appears in input to the TS the BUNCH2 tracks are masked, i.e. are ignored during the sorting. In the normal Cycle1-Cycle2 sequence, the TS selects the FBT and SBT. In case of two consecutive Cycle1, out of the data belonging to the first Cycle1 the TS selects the FBT of that cycle and could recover the next-to-best for transmission to the Regional Trigger. Alternatively, off-time tracks might be stored and used in the subsequent sorting cycle, or flagged so that the track finder in the Regional Trigger can associate them to some bunch (usually the previous one) of tracks. The masking logic is also used to deal with the H/L trigger priority;

- in the cascade processing approach sel and postsel lines are controlled by means of an additional simple logic. Each of the sel (or postsel) output lines from the TSP7 unit is used to enable driving of sel (or postsel) output lines from the corresponding TSP4 unit. In this way at most one sel (or postsel) line out of 28 is enabled at each BX. For each TC only a sel or a postsel line can be active at the same time and can enable the transmission of the complete best-track information. A sel line can enable the transmission of the track being processed at the current BX. Instead, a postsel line enables the transmission of a track calculated at the previous BX (i.e. belonging to BUNCH1), which must have been stored into a TC dummy register.
5 Conclusions

The Trigger Server of a muon station is part of the muon trigger chain of the DTBX chambers in the CMS barrel [2]. Every Track Correlator in a muon station can send to the TS up to two track-data which are ordered in quality. In total the TS should handle up to 56 tracks (28 tracks per BX in two consecutive BXs) and select the two best tracks for transmission to the Regional Trigger. The definition of best track must be consistent with the selection algorithm used in the TCs. The general requirements on the TS functionality have been studied:

- it is argued that sorting the true second-best track becomes particularly important in multi-muon events, for instance muonic Higgs decays. Then it is essential that the TS sorts best and next-to-best out of the 28 tracks of BUNCH1;

- three quality bits should accompany the $\Delta\psi_R$ value of a TC track: a H/L trigger bit, a first/second bunch bit, an inner/outer SL bit (or a correlation bit). These are necessary to maintain the TS selection algorithm consistent with the TC selection algorithm and to properly handle off-time and overlap triggers.

Two approaches to tackle the problem of sorting best and next-to-best tracks of BUNCH1 have been investigated in detail:

- a fast min-search algorithm is applied twice over 28 data words;

- a parallel min and next-to-min search algorithm is applied twice, in cascade over small groups of data words.

The sequential approach assumes that the selection of the best track is very fast, so that it is possible to run twice the selection for the next-to-best search. A bit-serial algorithm with 5-bit input words is used twice in sequence, involving a total of 40 levels of logic gates.

In the parallel approach the 28 input words are divided in 7 groups of 4 words and the analysis is performed by specific processing units, embedded in a tree structure of two layers. Best and next-to-best tracks are searched for in parallel. The parallel algorithm involves 21 levels of logic gates. We conclude that, on the basis of the response time, the parallel approach should be preferred. For an ASIC chip, according to the VERILOG simulation, the parallel sorting circuit can be executed in 11 nsec for each bunch of tracks. Therefore, the developed sorting algorithm fulfills the constraint of 25 nsec response time, and leaves enough time for I/O buffering, synchronization and control.

Acknowledgements

We are indebted to Dr. A. Gabrielli, Prof. E. Gandolfi and Prof. M. Masetti of the University of Bologna, and to Dr. M. De Giorgi, Dr. R. Martinelli and Dr. A.J. Ponte Sancho of INFN and University of Padova.
References


<table>
<thead>
<tr>
<th>Delay from BT1 trigger</th>
<th>TC trigger sequence</th>
<th>TSΦ trigger sequence</th>
</tr>
</thead>
</table>
| 1BX                    | read ψ and x from BTIs  
select lowest ΔψR: BTIi1 and BTIi1
(from all BTIs preferring H triggers) |                  |
| 2BX                    | send ΔψR of first track to TSΦ  
receive TRGθ from TSθ  
select lowest ΔψR: BTIi2 and BTIi2
(from all BTIs, but BTIi1,BTIi1) | receive ΔψR of first track from TC  
find first lowest ΔψR  
select first TC |
| 3BX                    | send ΔψR of second track to TSΦ  
calculate first track: TRK1  
set first CORRELLATION flag |                |
| 4BX                    | send TRK1 to TSΦ  
calculate second track: TRK2  
set second CORRELLATION flag | receive ΔψR of second track from TC  
find second lowest ΔψR  
select second TC |
| 5BX                    | send TRK2 to TSΦ | read first track from TC |
| 6BX                    | read second track from TC | |
| 7BX                    | transmit first and second tracks | |

Table 1: TC and TS (in the RΦ view) operational sequences [2].
Figure 1: Drift tubes of a BTI (shaded) in the outer SL and drift tubes of a BTI (shaded) in the inner SL along the trajectory of a muon in a muon station.

Figure 2: Correspondence between the TCs of a muon station and the BTIs of the station inner and the outer SL.
Figure 3: Trigger Server position in the trigger chain of a DTBX Muon Station.

Table 2: Example of track transmission sequence from TCs to TS. Symbols “1” and “2” indicate BUNCH1 and BUNCH2 data words respectively. The symbol “-” indicates a null data word in a given TC. The sorting cycles are shown in the last raw.
Figure 4: A bit-serial algorithm for min search.
Figure 5: Tree structure for parallel sorting of min and next-to-min, applied to 28 input words.
Figure 6: A TSP unit for 4 input words.