Available on CMS information server

CMS NOTE 1997/092

The Compact Muon Solenoid Experiment

CMS Note

Mailing address: CMS CERN, CH-1211 GENEVA 23, Switzerland

The Hardware Muon Trigger Track Finder Processor in CMS - Architecture and Algorithm

A. Kluge, T. Wildschek

CERN, Geneva

Institute of High Energy Physics, Vienna

Abstract

The paper discusses the architecture and functionality of the muon trigger track finder processor. The system segmentation and algorithm are described in detail. The system and algorithm has been optimized using the hardware description language VHDL. The processor is based on data from the drift tube muon chambers. The task of the processor is to identify muons and measure their transverse momentum p_t. Data of more than two hundred thousand drift cells are used to determine the location of muons and measure their transverse momentum.

INTRODUCTION

The basic architecture of the track finder processor is described. The mapping of the chamber structure onto the hardware level is discussed. Every single part of the processor model is discussed in detail. Due to the bending of the tracks and the non-projective geometry of the chamber system muons cross segment boundaries. This requires a large amount of interconnection between processing units. Using the hardware description language VHDL a simulation of the processor model was conducted. The model was used to prove the functionality of the algorithm. Moreover it served to optimize the system partitioning with respect to the amount of interconnections between processing units and processing latency. Using the VHDL model a FPGA prototype was designed [1,2,3,4,5].

LOGIC SEGMENTATION

Processing of the entire muon data of the detector within one logical unit is impossible and also unnecessary. The amount of 10 kbit data per crossing cycle yields an input data rate of about 400 Mb/s. The large amount of data to be processed causes a severe integration problem. When splitting up the processor in several physical units an interconnection problem between those units arises. Fortunately a muon track passes only a small number of detector segments [3].

In order to render communication between processing units possible at a minimum extent, the logical structure of the chamber system is mirrored inside the track finder processor hardware. In fig. 1 the logical segmentation of the track finder processor is shown.

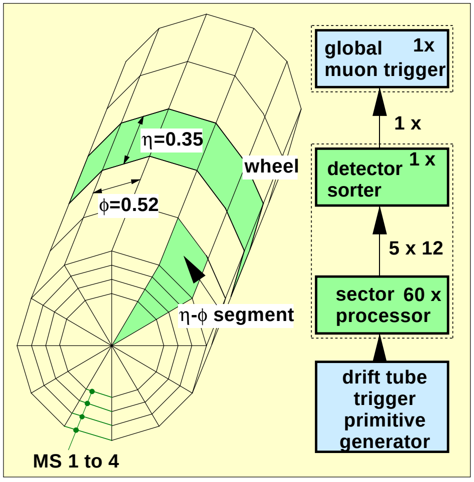

Fig. 1.: Logical segmentation of the track finder processor.

Like the detector itself it is divided into five wheels (Δz x Δϕ = 2.56 m x 2π). Each wheel is subdivided into twelve sector processors with a segmentation Δz x Δϕ = 2.56 m x 0.52 rad (30°).

A sector processor matches the track segments identified by the drift tube trigger primitive generator logic [7] and tries to form complete tracks. If the sector processor succeeds, it assigns a transverse momentum p_t, determines the location in ϕ and η to each track. Tracks which traverse more than one detector segment are given out by the sector processor of the detector segment from where the track's innermost track segment is found.

Of the 60 (12 ϕ times 5 wheels) times two possible tracks identified by the sector processors only the four tracks with the highest p_t are retained by the wheel sorter. All the information on these tracks - p_t, charge, η, ϕ, quality - is forwarded to the global muon trigger. The latter combines the track finder processor information with the trigger information given by the RPC-system [6, 8].

TRACK FINDER PROCESSOR ALGORITHM

In fig. 2 a simplified block diagram of a sector processor is displayed. The sector processor is divided into three parts - the extrapolator (EU), the track assembler (TA), and the p_t-, η-, ϕ-and quality–assignment units (AU).

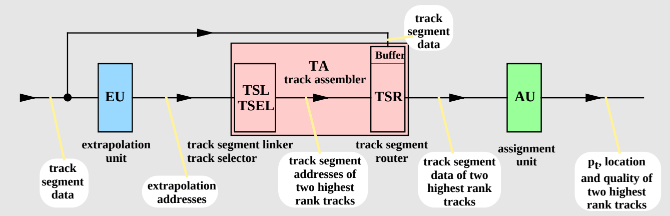

Fig. 2.: Block diagram of a sector processor.

The extrapolation unit EU attempts to match track segment pairs of distinct stations using the extrapolation criteria described in [1]. When track segment pairs meet these criteria the information is forwarded to the track assembler TA. All extrapolations between all station pairs are carried out in parallel.

Since tracks may cross detector segment boundaries the information of the extrapolation units of the neighbouring detector segments are also routed to the track assembler TA. The track segment linker (TSL) and track selector (TSEL) evaluate all extrapolation results in order to find up to two tracks with the innermost track segment in its own detector segment. They forward the relative track segment addresses of the track segments of found tracks to the track segment router TSR. During the execution of the track assembler algorithm the track segment data are stored in a buffer memory located in the TSR. The relative addresses are used by the track segment router TSR to extract the corresponding track segment data out of the buffer memory.

The track segment data are forwarded to the p_t-, η-, ϕ- and quality-assignment units (PAU, ηAU, ϕAU, qAU).

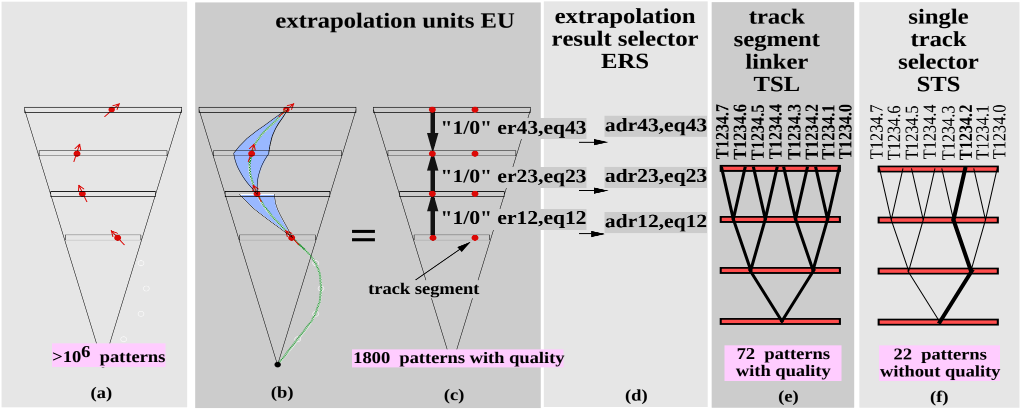

Fig. 4.: Reduction of possible track candidates by the track finder algorithm.

Short description of the track finder algorithm

In the following a short overview of the track finder algorithm is given. The various processing units are described in detail later. Fig. 3 shows a more detailed block diagram. The algorithm reduces the number of possible track candidates from about 1800 to two. Fig. 4 illustrates the first part of the reduction.

The extrapolation units (EU) match track segment pairs to each other. As a result the information bit *er* indicating which track segments belong to each other and the quality word *eq* of the matched track segments are given out. After the extrapolation 1800 possibilities to assemble valid track candidates exist. The track candidates are called track segment patterns. (fig. 4 b, c) .

The extrapolation result selector (ERS) selects the two best extrapolations for each source track segment. It outputs the relative address *adr* of the track segments as well as the extrapolation quality *eq* (fig. 4 d).

The track segment linker (TSL) attempts to link track segments together starting from the innermost track segment. As the extrapolation result selector (ERS) delivers up to two extrapolation addresses per source track segment more than one track candidate may be found originating from the innermost track segment (fig. 4 e). In order to cope with inefficiencies of the chamber system a given number of track segment linker modules start in stations other than station one. The track segment linking scheme reduces the number of track candidates from 1800 to 72. A quality information remains attached to each track candidate.

For each innermost source track segment the single track selector (STS) retains only the track candidate with the highest extrapolation quality. A total of 22 track candidates can survive (fig. 4 f).

The cancel out units (COL) cancel tracks using track segments already contained by longer tracks. Thus the cancel out units (COL) also erases track patterns which are part of longer track patterns. An example is a track consisting of track segments in station two and three which are found to be equivalent to track segment two and three of a track containing segments from all four stations.

The track class selector (TCS) selects the two highest ranking tracks out of the remaining 22 track candidates and forwards the relative addresses of the matched track segments.

The track segment router (TSR) uses these relative addresses mentioned above to output the corresponding track segment data.

The assignment units (AU) use the track segment data to determine the track properties.

In the following chapters all elements of the track finder processor are described in full detail.

EXTRAPOLATION UNIT (EU)

The extrapolation unit attempts to join a track segment with track segments of other stations. The extrapolation unit assigns to each possible track segment pair an extrapolation result and an extrapolation quality.

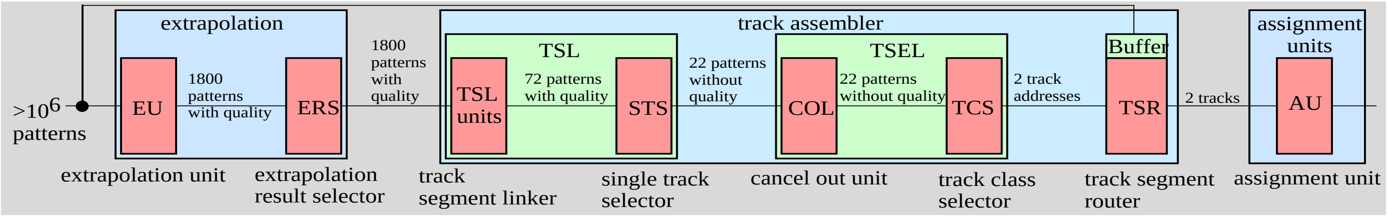

Fig. 3.: Block diagram of a sector processor.

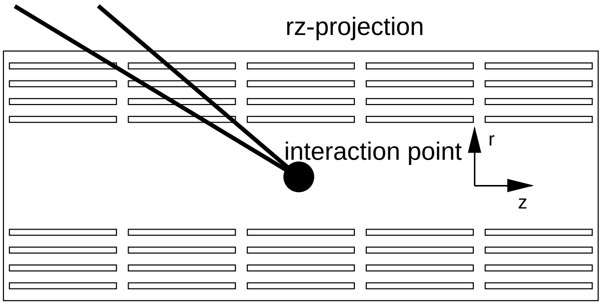

Fig. 5.: As seen in the rz-view muons can only traverse two wheels.

Due to the bending a muon can cross detector segment boundaries in ϕ-direction. Thus when joining two track segments it is necessary to involve the track segments of the neighbouring ϕ-sectors. The maximum deflection for strongly bent low p_t muons is found to be below the dimensions of two detector segments (2 · 0.52 rad or 2 · 30°) [3]. Fig. 6 shows the deflection of muon tracks between station one and two (ϕ_{21}) over their transverse momentum p_t for η=0..

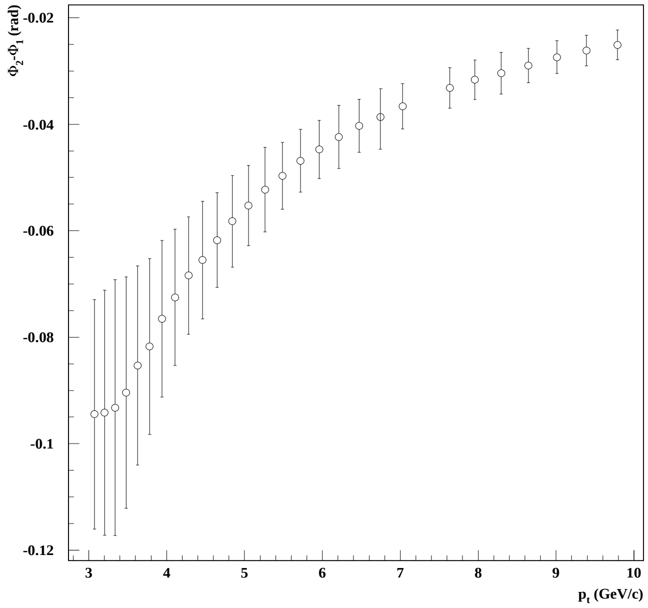

Fig. 6.: Deflection of muons between stations one and two ϕ_{21} over their transverse momentum p_t for pseudo rapidity η=0. The error bars give the ±σ deviation from the medium value.

The non-projective geometry of the chambers with respect to the muon tracks in the rz-view requires to examine also the neighbouring wheels. Fig. 5 illustrates that a muon originating from the interaction point does not cross more than one wheel boundary. The deviation from a linear track in the rz-plane is small. The track can be approximated by a straight line.

When joining track segments it is sufficient to look into the corresponding detector segment and its directly adjacent neighbours. As the muon track in the rz-projection is almost a straight line the particles will not change their flight direction with respect to the z-direction of the detector. Checking the wheel contrary to the flight direction is not necessary. In all wheels but the central wheel the flight direction of the muons in z-direction obviously is outgoing. Thus in all wheels but the central wheel the sector processor algorithm examines six detector ($\eta\phi$)-segments (fig. 7). When extrapolating from the centre of the detector (wheel 0) the target track segments of the nine adjacent detector segments have to be evaluated.

From now on the sector processor of an outer wheel is described. For the sector processor in the centre of the detector (wheel 0) the number of target track segments and thus the number of extrapolations has to be increased accordingly.

In every chamber the trigger primitive generator delivers up to two track segments. The extrapolation unit has to attempt to match each source track segment to twelve (twelve = two track segments per chamber times six neighbouring chambers) track segments of the next station (fig. 7)

Simulation of trigger acceptance shows that it is necessary to accept tracks with at least two out of four track segments [1]. Thus for the matching process six station pairings are necessary; 1-2, 1-3, 1-4, 2-3, 2-4 and 4-3. (The extrapolation from station three is not possible. The extrapolation is carried out the other way round [1].) In total twelve source track segments exist per detector segment. Thus the sector processor incorporates twelve separate extrapolation units. Each extrapolation unit extrapolates one source track segment to another station and compares the extrapolated value to twelve target track segments. As a consequence 144 comparisons are carried out in the processor.

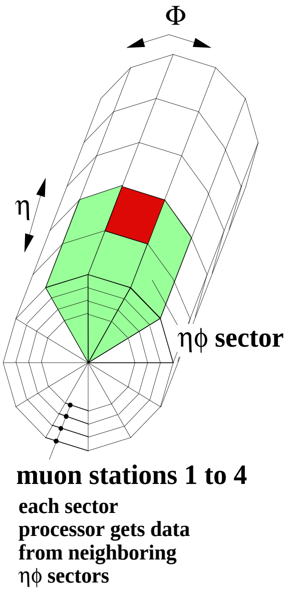

Fig. 7.: At least five neighbouring detector segments must be evaluated.

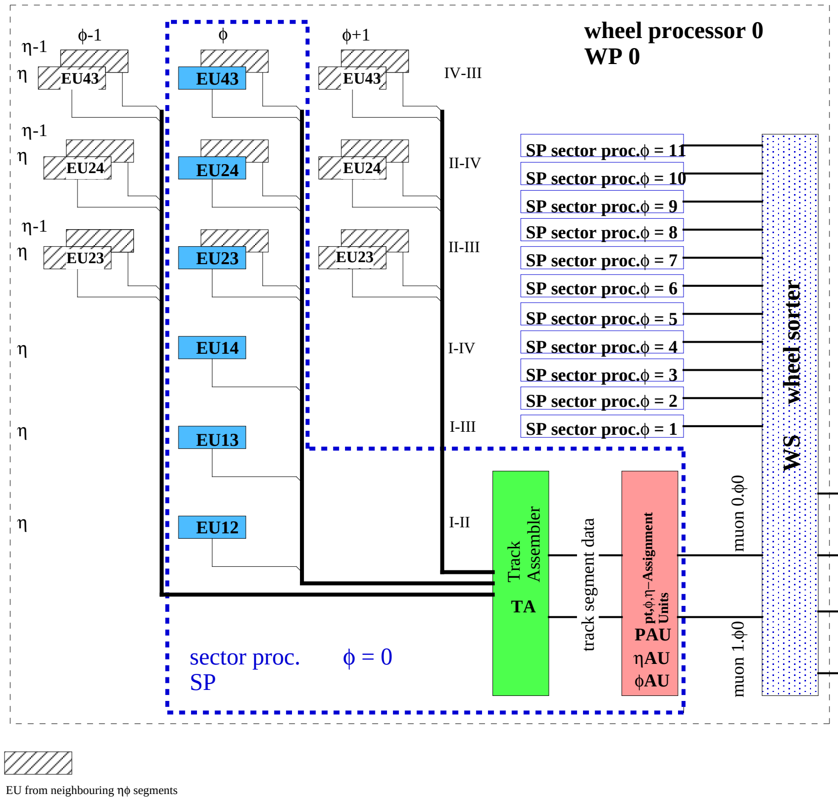

Fig. 8.: Block diagram of the entire track finder processor.

In order to process tracks crossing detector segment boundaries the information of neighbouring extrapolation systems must be made available to the sector processor. The output of the neighbouring extrapolation units EU23, EU24 and EU43 from the five neighbouring detector segments are routed to the sector processor. A sector processor searches for tracks with the first track segment within its detector segment. The output of the neighbouring extrapolation units EU12, EU13 and EU14 is not needed. The block diagram of the entire track finder system is illustrated in fig. 8.

Using the extrapolation method allows to calculate the expected deviation $\phi_{deviation}$ (difference between extrapolated hit position ϕ_{extra} and source hit position ϕ_{source}, see fig. 9) and an extrapolation threshold $threshold_{ext}$. The extrapolated hit position ϕ_{extra} is compared to the hit position of each possible target track segment ϕ_{target}. If the difference is found to be below the extrapolation threshold $threshold_{ext}$ the extrapolation is considered successful. The extrapolation result bit is set. An extrapolation quality word is assigned to each successful extrapolation; it is derived from the track segment qualities given by the drift tube trigger primitive generator.

The hardware implementation of the extrapolation using the medium deflection $\phi_{deviation}$ and the extrapolation threshold $threshold_{ext}$ proves cumbersome because the absolute value of the difference of the extrapolated hit position ϕ_{extra} and the actual hit position ϕ_{target} needs to be calculated. A window comparator is more efficient. It checks for the difference *diff* between hit position in the source chamber ϕ_{source} and the hit position in the target chamber ϕ_{target} to be within a lower and an upper limit, $diff^+$ and $diff^-$, which greatly reduces the number of calculation steps (see fig. 9).

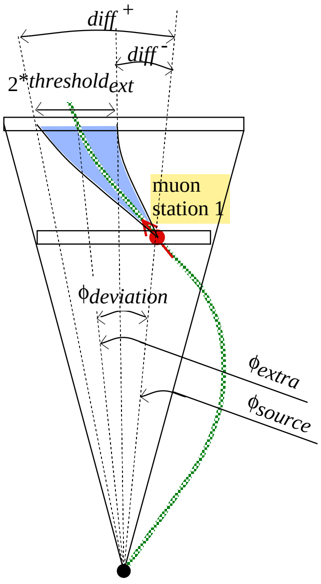

Fig. 9.: Extrapolation from station one to station two.

It is too time consuming to obtain the extrapolation limits by means of digital arithmetic logic units. The values $diff^+$ and $diff^-$

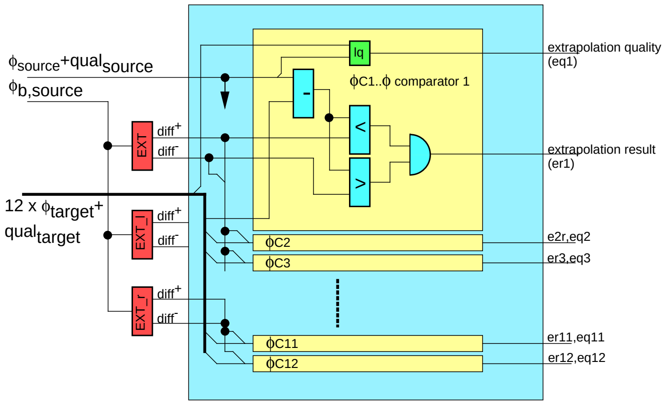

Fig. 10.: Extrapolation unit extrapolates from one track segment and compares the extrapolated value to twelve target track segments. It outputs the extrapolation result (er) and the extrapolation quality (eq) for each comparison.

are therefore predetermined and stored in memory based lookup tables (EXT). The bend angle ϕ_b of the source track segment is used to address two RAM-based lookup tables which output the upper and the lower limit (*diff*$^+$ and *diff*$^-$) of the difference of the position values of the two track segments in question. At the same time the actual difference *diff* between the hit position in the source chamber ϕ_{source} and the hit position in the target chamber ϕ_{target} is calculated. A window comparator checks if the difference *diff* is found to be within the given limits (*diff*$^+$ and *diff*$^-$). In case of a successful extrapolation the extrapolation result bit (*er)* and the extrapolation quality word (*eq)* is assigned. In the FPGA prototype [2,3] the extrapolation quality is a one bit word (it is set to ‘1’ if both track segments of the matched pair have the quality status ‘correlated’ ; i.e. a track segment uses measurements from both rϕ-superlayers [7]). For the final implementation further simulations have to show whether this single quality bit is sufficient. Each extrapolation unit outputs the extrapolation results and the extrapolation quality information for each of the twelve target track segments (see fig. 10).

A problem arises when subtracting the hit coordinates ϕ_{source} and ϕ_{target} of track segments of different ϕ-segments. The origins of coordinates are different and thus the difference *diff* is shifted by the chamber dimension in ϕ (0.57 rad). The obvious solution were to add or subtract this offset from the target track segment position ϕ_{target}. The disadvantage is the additional calculation delay. In order to avoid this delay the offset compensation is carried out already during the calculation of the extrapolation limits *diff+* and *diff-*. Accordingly two further extrapolation lookup tables must be introduced; one is used for the extrapolations to ϕ-sectors on the right hand side (EXT_r) and the other one for the left hand side (EXT_l). The lookup tables EXT_l and EXT_r deliver the offset-compensated extrapolation limits *diff*$^+$and *diff*$^-$(fig. 10).

Extrapolation result selector (ERS)

The extrapolation result selector selects the two best extrapolations per source track segment and outputs the relative address of the target track segment.

The selection criteria are the extrapolation qualities and the relative location of the target track segment with respect to the source track segment. The first criterion is the extrapolation quality. In case of equality the target track segment is checked for its origin which may be either the same detector segment or a neighbouring segment.

For each of the two target track segments the extrapolation result selector ERS outputs:

- Relative address of the target track segment. In case the extrapolation was unsuccessful a certain code is given out.
- Extrapolation quality. The extrapolation quality the selection was based on.

The hardware implementation uses a set of priority encoders.

The main task of the extrapolation result selector is to reduce the data stream from the extrapolation units to the track assembler. The probability to find more than two successful extrapolations originating from one single track segment is negligibly small. However, the extrapolation units deliver the extrapolation result and the extrapolation quality for each of the twelve possible track segment pairs. The extrapolation result selector simplifies track segment assembly.

TRACK ASSEMBLER (TA)

The task of the track assembler is to find the two tracks in a detector segment exhibiting the highest number of matching track segments and the highest extrapolation quality. It outputs the track segment measurements belonging to these tracks. The track assembler TA consists of the track segment linker TSL, the track selector TSEL and the track segment router TSR.

The task of track segment linker is to link the track segment pairs formed by the extrapolation units to full tracks. It forwards all track candidates to the track selector, which selects the two highest ranking tracks and gives out the relative addresses of the matching track segments. The track segment router extracts the corresponding track segment data from the data pipeline using the relative addresses of the track segments.

Track segment linker (TSL)

A pattern matching method may be considered to implement the track segment linker. The output of the extrapolations units (extrapolation result and quality) can be used. A valid track consisting of *n* track segments is identified by a pattern of *n-1* extrapolation result bits. This task can be performed using simple gate array logic. Each valid track segment combination is recognized by its dedicated AND gate. A priority encoder (pattern selector) retains only the two highest ranking patterns and gives out their code. This code unambiguously identifies the track segments contained by the found tracks. Fig. 11 shows examples of valid track segment pair patterns. In total there are about 1800 valid patterns. On one hand this relatively high number of valid patterns is due to particles crossing detector segment boundaries. On the other hand this high number arises due to possible tracks consisting of less than four track segments. Even a single track segment pair has to be accepted as a full track. There are eleven valid track classes overall: T1234, T123, T124, T134, T234, T12, T13, T14, T23, T24 and T34 (the digits denote the stations belonging to the track).

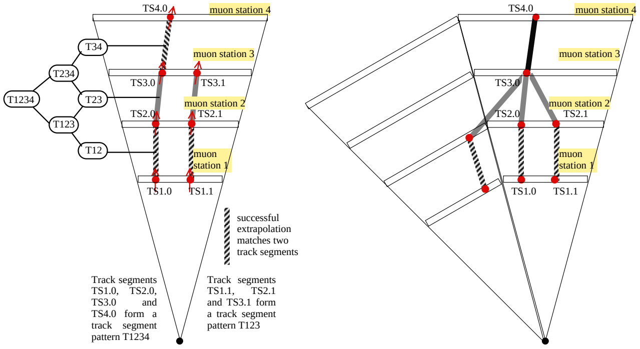
TS4.0
muon station 4
T34
muon station 3
T234
TS3.0
TS3.1
T1234
T23
muon station 2
TS2.0
TS2.1
T123
T12
muon station 1
TS1.0
TS1.1
successful extrapolation matches two track segments
Track segments TS1.0, TS2.0, TS3.0 and TS4.0 form a track segment pattern T1234
Track segments TS1.1, TS2.1 and TS3.1 form a track segment pattern T123
TS4.0
muon station 4
muon station 3
TS3.0
TS2.0
muon station 2
TS2.1
muon station 1
TS1.0
TS1.1

Fig. 11.: Example of track linker patterns.

A track consisting of four matching track segments, T1234, is assembled by three matching track segment pairs (1-2/2-3/3-4). However, both the combination of track segment pairs '1-2/2-3' and '2-3/3-4' each form an additional valid track (T123 and T234) consisting of three matching track segments. Furthermore even each single track segment pair (1-2,2-3,3-4) forms more valid track segment pair patterns. This is illustrated in fig. 11. These sub patterns must be suppressed in case of a complete track T1234. Basically the recognition of 1800 patterns of a length of up to four bits is easily configured by hardware. Problems arise when the sub patterns need to be suppressed. Every single 'short track' (consisting of only two or one track segment pairs) is a sub pattern of a vast number of longer tracks (see fig. 11). This fact raises a severe interconnection and integration problem within the track segment linker. In order to find tracks overlapping across detector segment boundaries the extrapolation results of the detector segment neighbours also have to be evaluated. Sub patterns have to be cancelled out even by tracks originating from a different detector segment.

The extrapolation qualities must be taken into consideration in order to find the best track. It is not sufficient to recognize the track pattern and to mark its appearance. Furthermore the attached quality information must be provided to the pattern selector. The latter must select the highest ranking pattern with respect to the number of involved track segments and with respect to extrapolation quality. This process is time consuming and complicated because the pattern selector would have to compare quality words of 1800 patterns to each other.

Considering these requirements the hardware implementation using a pattern matching method for the track segment linker proves impossible. Each short pattern might be vetoed by a longer track. Simple four fold AND gates to recognize full patterns are by far not sufficient.

Applying the extrapolation method reduced the number of patterns to some 1800. Even at this stage a pattern comparison method in the track segment linker is not feasible.

Dynamic track segment linking with indirect addressing

The track segment linker comprises the track segment linker units and the single track selectors. Each of the track segment linker units is responsible to find tracks of a certain track class originating from one track segment. The track segment linker units forward their track candidates to their single track selectors. There is one single track selector for each track segment linker unit. The single track selectors retain only the track candidate with the highest extrapolation qualities. The linking result and the relative track segment addresses of all involved track segments for each track class and source track segment are given out.

Since the scheme works comparable to the indirect addressing in a microprocessor and links the track segment pairs without comparing it to a predefined set of patterns it is called 'dynamic track segment linking with indirect addressing'. The basic principle is to assemble all tracks from their innermost starting points in a serial way.

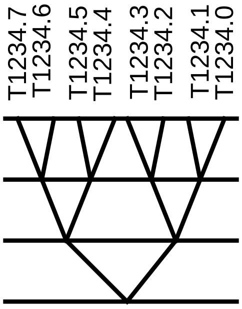

Fig. 12.: Eight track candidates of the track class T1234 emerge from a single track segment.

The extrapolation result selector delivers the addresses of the target track segments of previously formed track segment pairs. In order to append a second track segment pair to form a track consisting of three track segments one has to look if an extrapolation from the target track segment of the first track segment pair to another track segment was successful. One looks at the output of the extrapolation result selector of the first track segment pair. The obtained extrapolation address denotes the source track segment for the track segment pair to append. If the extrapolation result selector of this source track segment gives a valid target segment address, this address indicates the address of the linked track segment. A track consisting of three track segments is formed. The scheme is repeated once more in order to form tracks comprising four track segments.

One track segment linker unit attempts to find a track candidate for one track class originating from one single track segment. There are eleven track classes. Two track segments in each station are delivered by the drift tube trigger primitive generator. As a consequence (2 · 11 =) 22 track segment linker units are being applied and form one track segment linker.

Every extrapolation result selector ERS delivers up to two addresses. The theoretically possible number of track branches originating from the same track segment is 2^{n-1}. n is the number of involved track segments (see fig. 12). The sum of all valid track branches yields 72. For every possible track branch all track segment addresses are given out. The linking result, indicating the presence of a track candidate, can be derived from the track segment addresses of the outermost station. If it equals the code for a valid track segment, this track candidate has been “found”.

The result of this scheme is a priority ordered bit stream of 72 bits. The dynamic track linking scheme reduces the number of patterns from about 1800 to 72 by a factor of 25 without sacrificing measurement accuracy.

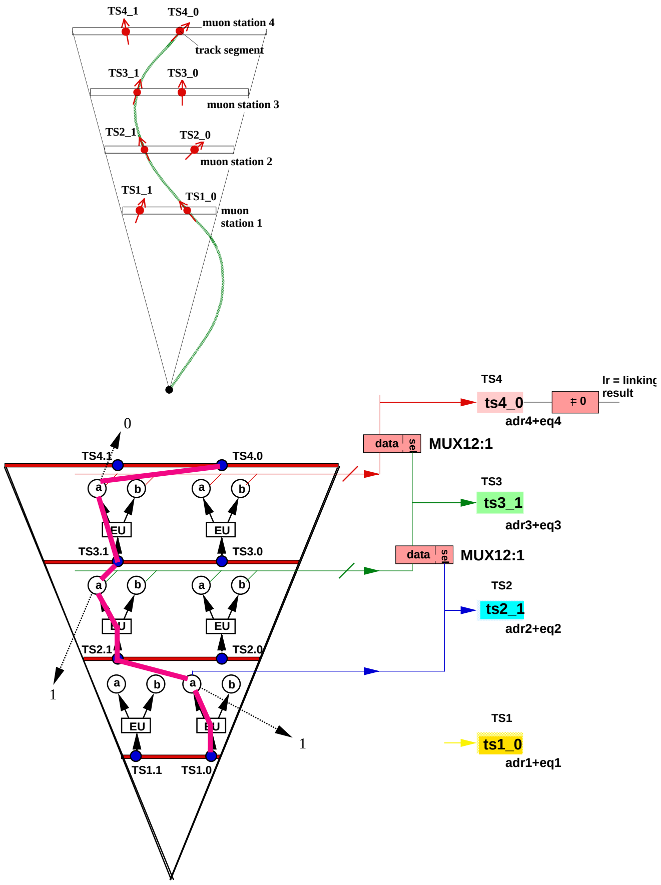

Fig. 13.:Principle of track segment linker operation.

An example is given in fig. 13. It assumes a track to be comprised by track segment zero in station one, track segment 1 in station two and three and track segment zero in station four. As a consequence the extrapolation units and extrapolation result selectors of the extrapolation from track segment zero in station one to station two give the address of track segment on (in station two). The extrapolation units of track segment on in station two give out the address for track segment one1 in station three and so on.

As illustrated in the figure the extrapolation addresses *a* guide the track linker unit from the innermost track segment to the outermost.

The hardware implementation employs multiplexers (see fig. 13). The extrapolation addresses of the first track segment pair are connected to the select input of a multiplexer. All target addresses of extrapolations of the next station pair are routed to the data input. The address on the select inputs selects the according target track segment address among twelve. Thus a twelve to one multiplexer is employed. The output of the multiplexer is the address of the appended track segment. This principle resembles the indirect addressing in a microprocessor. The advantage of this method lies in the speed of multiplexers implemented in hardware. Only a small number of logic gates is required. Although the scheme proceeds in a serial way (with up to two multiplexer stages in a row) it is much faster than a pattern comparison approach. An important issue for the inevitable error monitoring is the clear and simple concept of the scheme.

Single track selector (STS)

The single track selector retains only one track originating from one source track segment by selecting the track with the highest extrapolation quality.

For two reasons only one track originating from the same track segment is tolerated. Firstly multiple hits in the drift chamber caused by δ-rays will trigger several valid track candidates (see fig. 14). Secondly it is time consuming to select the two best patterns amongst 72 if the extrapolation quality has to be included in the selection process.

Anyway, it is very unlikely to find a track segment measurement caused by two different muons but with exactly the same location. Thus the ‘single track selection’ STS may safely be applied. The single track selector retains one track branch per source track segment.

Fig. 14 illustrates the principle of the single track selection. δ-rays triggered track segment measurements in station two and four. Four track candidates are being provided by the track segment linker units TSL1234. The single track selector cancels branches with lower extrapolation quality *eq* and retains only the track consisting of track segments with the highest quality attached to.

Simulation showed that in most cases track segments caused by δ-rays have an uncorrelated track segment quality status [9]. This is due to the limited range of the electrons in matter. Hence track segment pairs with a track segment caused by δ-rays are likely to have low extrapolation quality.

Starting from the outermost branching point (in case of the tracks T1234 this is the extrapolation between station 3 and 4) the branch with the lower extrapolation quality is cancelled. In case of equality the branch with the higher internal index is

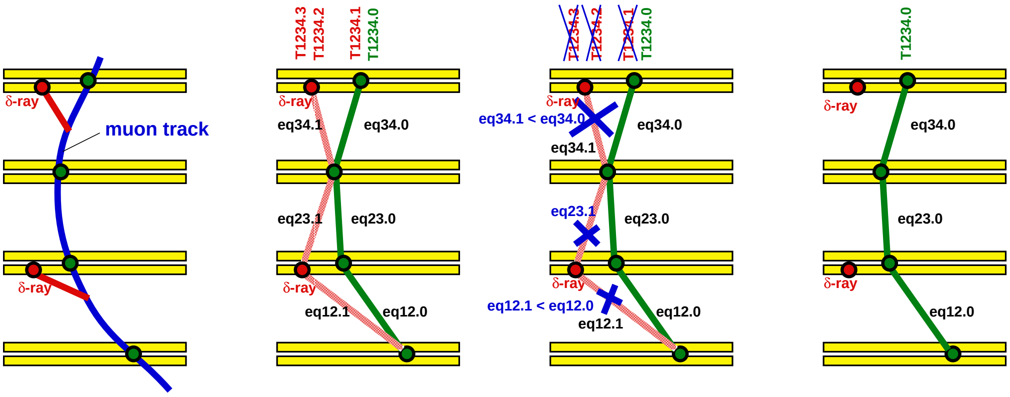

Fig. 14.: Principle of single track selection operation.

removed. After selecting the highest ranking branch 34 the single track selection is applied again to the branches 23 and 12.

The number of possible tracks is now reduced to 22. Only one track per track origin is given out. Applying the single track selection reduces the number of valid patterns from 72 to 22.

The hardware implementation employs multiplexers and comparators (see fig. 15). The extrapolation qualities of the branches are compared. The comparison result controls the multiplexer. It routes the address and the linking results of the remaining branch through the multiplexer.

All remaining linking results - a 22 (two start track segments times eleven track classes) bit field with each bit indicating if the corresponding track segment combination was found - and the track segment addresses are forwarded to the track selector. However, there is no quality information attached to the track candidates anymore. The single track selector already includes the quality information into its decision.

Simulation showed that for up to 75 % of muon tracks at least one station delivers a second track segment [10]. Assembling the tracks without the single track selection scheme would cause the track finder processor to output a second track in all these cases. VHDL simulation of the processor revealed that application of the single track selector decreased the fraction of double tracks given out to only 4 %. However, optimization of the drift tube trigger primitive generator are ongoing in order to reduce the number of double or ghost track segments delivered to the track finder. However, this reduction is performed without any loss of real tracks. Certainly algorithms may be found which reduce the number of double tracks further but this might be at the cost of efficiency when two real tracks close to each other need to be treated.

Track selector (TSEL)

The track selector retains only the two highest ranking track candidates amongst the 22 provided. It consists of the cancellation units and the track class selector.

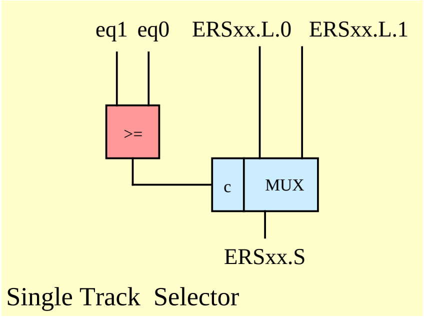

Fig. 15.: Block diagram of a single track selector.

The previous stage, the track segment linker, provides all possible track candidates. However, some of the track candidates may be entirely part of a longer track candidate. For instance this is the case if a track containing track segments from all four stations, T1234, is being found. In addition tracks containing some of the same track segments (like T123, T234, T12,...) will be found. The track with the highest number of track segments is referred to as mother pattern while the others are called sub patterns. The cancellation unit suppresses the sub patterns in presence of their mother patterns and removes tracks of different track class originating from one track segment. The inputs are the 22 linking result bits from the track segment linker. The linking result bit of a certain track class is masked off if this track is part of a longer track. The cancellation unit outputs the masked 22 linking result bits to the track class selector, which selects the two highest ranking track candidates and forwards the relative address of the track segments.

Cancellation units (COL)

The basic principle of the cancellation scheme is to alter only the linking result bits in presence of a mother pattern. The track segment addresses remain unchanged. The hardware effort is minimized. The cancellation units are divided into three groups: cancellation sub pattern, cancellation single track and cancellation down.

Cancellation sub pattern checks the presence of a mother pattern. Cancellation single track attempts to distinguish between multiple tracks caused by δ-rays or by tracks close together and removes tracks caused by δ-rays. Cancellation down finds double tracks caused by double hits in the innermost chamber.

The hardware implementation employs simple logic gates and comparators.

Track class selector (TCS)

The task of the track class selector is to find the highest ranking track amongst all track candidates. On the set of track classes (as defined above) an ordering is defined. The first criterion is the number of track segments the track consists of. The second criterion prefers tracks with track segment of stations one and two, because the momentum can be measured with a higher precision in station one and two [10].

Applying the extrapolation method, the dynamic linking of track segment pairs and the single track selection reduces the number of track candidates to 22. No quality information must be taken into account anymore at this stage. Thus a simple priority encoder may be used. The linking result bits are connected to the data input in the order of their rank. The track class selector puts out the relative addresses of the track segments of up to two tracks and forwards them to the track segment router TSR.

TRACK SEGMENT ROUTER (TSR)

The track segment router uses the track segment addresses to select the corresponding track segment data. During the processing time of the track assembler the data are stored in a buffer memory. Shift registers and multiplexers are used.

ASSIGNMENT UNITS (AU)

Once the track segment data are available to the assignment units, memory based look up tables are used to determine the transverse momenta of the particles. The momentum and the location of the tracks as well as a quality information about the track finding is given out.

P_t-assignment unit (PAU)

The resolution for determining p_t is dependent on the track segment quality. A control circuit evaluates the various track segment qualities and determines the p_t measurement algorithm [10]. In case transverse momentum p_t is measured using two spatial coordinates, the difference between the position values is sent to the memory based p_t assignment look up tables. If p_t is assigned using only an angle of a track segment this angle is routed to the look up table. p_t is measured with a resolution of 5 bit, one additional bit indicates the charge of the particle.

ϕ-assignment unit (ϕAU)

Measurement in ϕ can be given out very accurately since the position of the track segments in ϕ is known well. The binning of the track segments corresponds to a ϕ-resolution of 0.018° or 0.3 mrad. That means the track position in ϕ can be given in the same resolution. The first level global trigger cannot process the sub trigger data with such a good resolution. The ϕ-value of the innermost track segment of the track is given out with a reduced resolution of 8 bits for the whole 2π range. This is equivalent to a binning of 1.4° or 25 mrad.

η-assignment unit (ηAU)

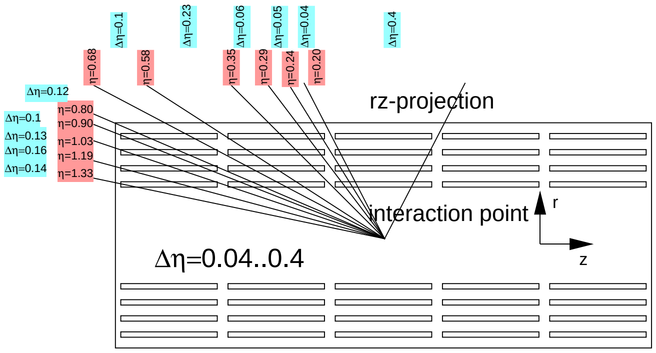

Fig. 16.: rz-view of the drift tube chamber system.

The non-projective chamber geometry with respect to the rz-view does not allow for a precise measurement of the η-coordinate of the track. As illustrated in fig. 16 the only information the η-coordinate can be derived from is the place where a particle crossed detector wheel boundaries. A coarse, location dependent resolution in η can be achieved. Moreover the resolution is dependent on the number of track segments a track consists of. The η-measurement uses a 2 bit code which indicates between which stations the muon crossed the wheel boundaries.

Quality assignment unit (qAU)

A quality code for each track is given out. One bit indicates if the transverse momentum has been assigned using the difference of two spatial measurements or by using one track segment angle only. The p_t measurement using two spatial coordinates yields a measurement with a better resolution. Further two bits are reserved to output the number of track segments involved in the track. Since a valid track may consist of two track segments only the number of track segments a track contains represents a quality measure. However, the quality information described here serves only as suggestion. Further global trigger simulation studies must provide detailed specification for the quality assignment unit.

WHEEL SORTER (WS)

The task of the wheel sorter is to select the four muons with the highest p_t in a wheel amongst the 24 muons from the twelve sector processors. This is done using a dedicated ASIC-sorter chip [11]. The latency for sorting four muons amongst 24 is 150 ns or 6 bunch crossings (bx). One bunch crossing is reserved for resynchronisation.

CONCLUSION: PROCESSOR ARCHITECTURE AND SIMULATION

The architecture was simulated using a behavioural VHDL simulation. The simulation was used to prove the functionality of the algorithm and to optimize the hardware partition.

The processor uses only simple logic blocks as such multiplexers, comparators and subtractors. Therefore the architecture proves to be simple and easy to maintain.

The efficiency curves show the performance of the entire track finder processor system (fig. 17) for p_t-thresholds p_t = 20, 40, and 50 GeV/c [10].

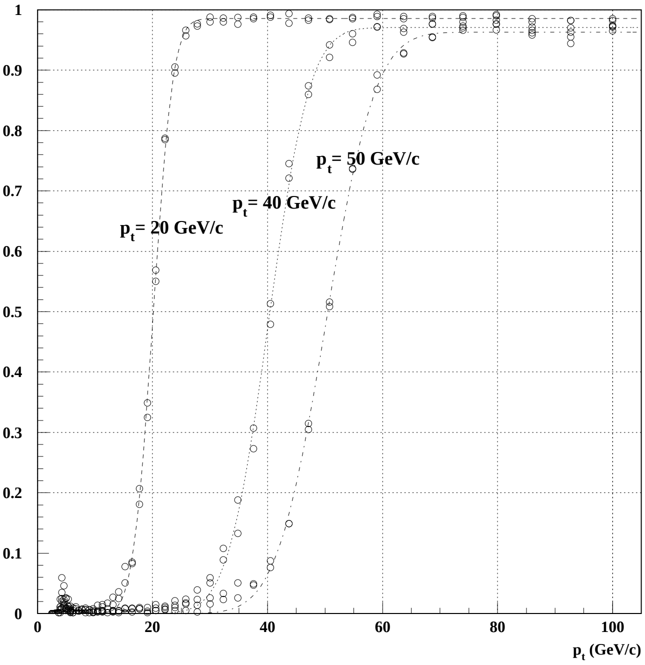

Fig. 17.: Efficiency curves for p_t = 20, 40 and 50 GeV/c.

The efficiency reaches at least 95 %. Not all muons are found because of geometrical acceptance of the detector and inefficiency of the chamber system.

The obvious bottleneck of the track finder processor is the limitation to be able to process only up to two successful extrapolations per source track segment (extrapolation result selector). However, as VHDL simulations showed, in only 2 % of the double muon events more than two successful extrapolations occurred. Again they were caused by δ-rays. The extrapolation result selector is optimized in a way to select the track segment pairs with the highest quality. The extrapolation result selector discards the pairs caused by δ-rays. In this way the extrapolation result selector does not compromise system performance.

REFERENCES

[1] A. Kluge, T. Wildschek, The Hardware Muon Track Finding Processor in CMS - Specification and Method, CMS Note 1997/091

[2] A. Kluge, T. Wildschek, The Hardware Muon Track Finding Processor in CMS - Prototype and Final Implementation, CMS Note 1997/093

[3] A. Kluge, The Hardware Track Finder Processor in CMS at CERN, Dissertation at the Technical University of Vienna, October 1997

[4] A. Kluge, T. Wildschek, Track Finding Processor in the DTBX Based CMS Barrel Muon Trigger, Second Workshop on Electronics for LHC Experiments, CERN/LHCC/96-39, October 21, 1996.

[5] A. Kluge, T. Wildschek, The Track Finder of the CMS First Level Muon Trigger, Third Workshop on Electronics for LHC Experiments, CERN/LHCC/96-39, October, 1997.

[6] CMS, The Compact Muon Solenoid, Technical Proposal, CERN/LHCC 94-38, LHCC/P1, 15 December 1994

[7] M. De Giorgi et al., Design and Simulations of the Trigger Electronics for the CMS Muon Barrel Chambers, CMS TN/95-01, CERN, 12 January 1995

[8] N. Neumeister et al., CMS Global Trigger, CMS TN/97-009, January 20, 1997.

[9] A. Kluge, T. Wildschek, Track Finding Processor in the DTBX Based CMS Barrel Muon Trigger, First Workshop on Electronics for LHC Experiments, CERN/LHCC/95-56, October 1, 1995.

[10] T. Wildschek, An algorithm for Track Finding in the CMS Muon Trigger, Dissertation, Technische Universität Wien, 1997.

[11] Robertis G. De and Ranieri A., The Sorting Processor Project, CMS TN/95-028, March 6, 1995.