Feasibility of the Hardware Muon Trigger Track Finder Processor in CMS

A. Kluge, T. Wildschek
CERN

ABSTRACT

This paper describes a feasibility study for the design of the Muon Trigger Track Finder Processor in the high-energy physics experiment CMS (Compact Muon Solenoid, planned for 2005) at CERN. It covers the specification, proposed method, and a prototype implementation. Comparison between several other measurement methods and the proposed one are carried out. The task of the processor is to identify muons and measure their transverse momenta and locations within 350 ns. It uses data from almost two hundred thousand detector cells of drift tube muon chambers. The processor searches for muon tracks originating from the interaction point by joining the track segments provided by the drift tube muon chamber electronics to full tracks. It assigns transverse momentum to each reconstructed track using the track's bend angle.

TRACK FINDER PROCESSOR ENVIRONMENT

The detector CMS [1] will work at the hadron collider LHC. Protons will collide with a centre of mass energy of 14 TeV. Every 25 ns a bunch crossing occurs.

The detector CMS will be built around a high-field superconducting solenoid leading to a compact design for the muon spectrometer, hence the name Compact Muon Solenoid (CMS). The solenoid has an inner radius of 3 m generating a uniform magnetic field of 4 T parallel to the beam axis. The magnetic flux is returned through a 1.8 m thick iron yoke instrumented with muon chambers. The magnetic field in the return yoke is 1.8 T. The overall dimensions of the detector are: a length of about 20 m and a diameter of 14 m.

The muon detector fulfils three basic tasks: muon identification, trigger, and momentum measurement. The muon detector is placed behind the calorimeters and the magnet coil. It consists of four muon stations interleaved with the iron return yoke plates. The stations are numbered from 1 to 4 from inside out. A system of drift tubes (DT) [2] is applied in the barrel region, while cathode strip chambers (CSC) cover the forward region. In addition resistive plate chambers (RPC) cover the entire muon detector [1]. Fig. 1 shows the rϕ-view of the detector and a muon traversing the tracker, the calorimeters, magnet coil and muon system.

The track finder processor described in this paper processes data from the drift tube system. The DT-trigger primitive generator (TPG) [2] first processes the information of the

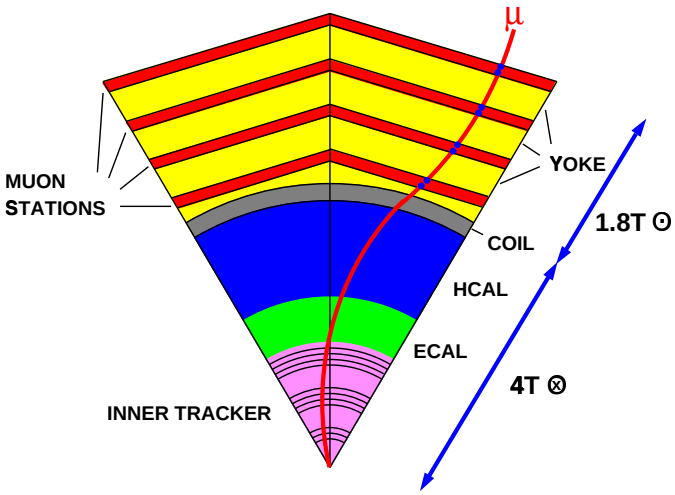

Fig. 1.: rϕ-view of the detector CMS and a muon track.

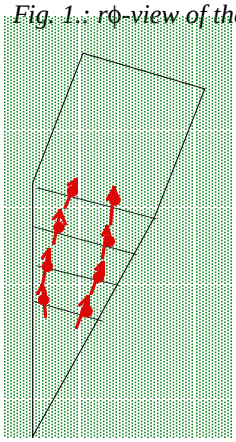

Fig. 2.a: Up to two track segments per chamber are given out.

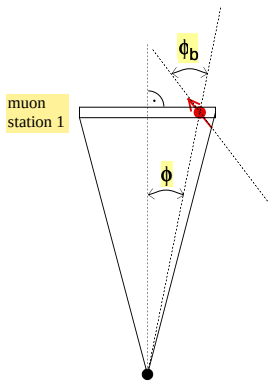

Fig. 2.b: A track segment consists of the spatial coordinate ϕ (11 bit), the bend angle ϕ_b (8 bit), and quality (3 bit).

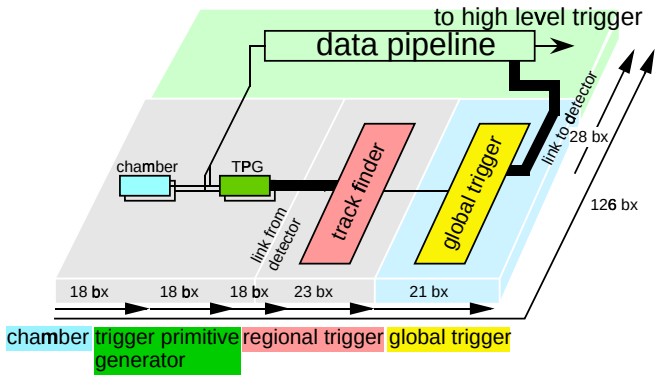

Fig. 3.: Muon trigger track finder processor data chain. Numbers give the latency of the components.

chamber locally. Up to two track segments (position and angle, see fig 2.a, b) per muon chamber are delivered.

The TPG provides a position resolution of 1.25 mm in station one and two and 2.5 mm in station three and four [3,4,5,6,7]. The choice of this input resolution to the track finder processor is motivated as follows: 1.25 mm is the finest resolution that can be provided by the TPG at reasonable expense. The track finder processor should have a fine momentum resolution at high momenta, where the complimentary RPC-based trigger suffers from a poor position resolution. The relevant quantity for momentum measurement is not the linear position, but rather the azimuthal angle. The choice of resolution given yields approximately constant angle resolution in the four stations (about 0.3 mrad).

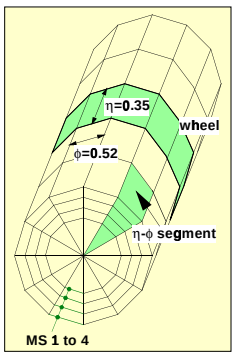

Fig. 4.: DT-chamber segmentation.

The DT-chamber system is divided into 12 ϕ-segments and 5 wheels in *z*-direction (see fig. 4). There are 4 muon stations. Thus the entire system comprises 240 chambers. Hence 480 track segments are delivered to the regional drift tube trigger, the track finder.

Track segments from different stations are collected by the track finder (see fig. 3). The task of the track finder processor is to find muon tracks originating from the interaction point and to measure their transverse momenta p_t and locations in azimuth ϕ and pseudo-rapidity η. The track finder selects the four highest p_t muons in the detector and forwards them to the global muon trigger.

TRACK FINDER PROCESSOR SPECIFICATIONS

In the following the main features of the track finder processors are listed and described shortly.

Output quantities

- transverse momentum p_t

Muons with a p_t below 2.0 GeV/c do not reach the chambers due to the bending in the magnetic field and energy loss in the absorbers. The p_t-range above 2.0 GeV/c is divided into 25 p_t-classes (see table 1). The p_t-value is given out in a 5 bit number. A sixth bit indicates the charge of the particle.

- location in ϕ

A ϕ-resolution of 1.4 or 25 mrad is provided by encoding the location in 8 bits.

- location in η

The η-coordinate can be derived from the place where a particle crossed detector wheel boundaries. The η-value is given in a 2 bit code. An option to improve η-resolution is to include trigger primitives of the middle (rz) drift tube layers, measuring the z-coordinate.

p_t range [GeV/c]	p_t code	p_t range [GeV/c]	p_t code	p_t range [GeV/c]	p_t code	p_t range [GeV/c]	p_t code
no muon	0	4-5	8	17-20	16	70-80	24
reserved	1	5-6	9	20-25	17	80-100	25
reserved	2	6-7	10	25-30	18	100-120	26
reserved	3	7-8	11	30-35	19	120-140	27
2-2.5	4	8-10	12	35-40	20	140-∞	28
2.5-3	5	10-12	13	40-50	21	reserved	29
3-3.5	6	12-14	14	50-60	22	reserved	30
3.5-4	7	14-17	15	60-70	23	reserved	31

Table 1: p_t-classes.

- Quality information

Quality information indicates the confidence that the found track is a real track and not a ghost track. Moreover, it gives the p_t-measurement resolution that can be expected from this track. Two 2-bit words are foreseen.

Requirements of the system

- Dead time free

The first level trigger architecture requires a dead time free operation. That means that even in case of a trigger accept by the global trigger the track finder processor must stay operational for the subsequent events. The trigger system is capable of accepting data from each single bunch crossing, with a data repetition rate of 40 MHz.

- Processing time - latency

It is important to keep the processing time of the track finder processor as low as possible, because during the time an event is evaluated in the level one trigger all corresponding detector data must be stored in the data pipeline. For the track finder processor 23 bunch crossings or 575 ns are reserved [8]. This number includes sorting of the four highest p_t muons. For the track finding and p_t-measurement only 350 ns are available.

- Programmability

The trigger system has to be flexible enough to permit changes in the algorithm and in the detector geometry. Even if the designed chamber geometry is not going to be changed, misalignment of the chambers must be accounted for.

- Output segmentation

The trigger system must output the information about four muons with the highest p_t in the detector.

- Technology

It must be possible to implement the hardware using today's technology.

Table 2 summarizes the track finder processor specifications.

track finder processor specifications
outputs the four highest p_t muons per detector: p_t, location, quality
p_t measurement range: 2.0 - ∞ GeV/c
ϕ-measurement: 25 mrad resolution
η-measurement: 0.04 - 0.4 resolution
dead time free
programmability: algorithm settings and chamber alignment
processing time: ≤ 23 bunch crossings or 575 ns.

Table 2: Track finder processor specifications.

IMPLEMENTATION OPTIONS

Several methods for the implementation of the track finder processor have been evaluated: Pattern comparison, histogram method, neural networks and extrapolation method. It should be pointed out that the assessment of the various technologies might be quite different by the time CMS starts up in 2005. However, for this feasibility study we have focused on what is achievable with today's technology, in accordance with the technology requirement listed in the previous chapter.

Template Matching (Pattern comparison)

The classical method in track finding hardware implementations is to search for predefined tracks or bit patterns. The actual hit patterns are compared to the predefined patterns.

Application of content addressable memories [9,10,11,12,13] is an obvious solution. However, it has to be stated that content addressable memories are available commercially only with small storage capacity. Examples can be found in [14,15]. The storage depth is of the order of 1024 x 48 bit words.

Simulations have been conducted to estimate the number of patterns to store. This number depends on the input resolution used, the bending of the tracks and the amount of multiple scattering. Due to the high magnetic field and large amount of material in the CMS muon system, the number of patterns exceeds $2*10^6$ [16], if one were to use the full resolution for storing the patterns. Full resolution is not needed for track finding, only for p_t-assignment. So one could use a two-stage design, where the first stage finds tracks, using a coarse position resolution, while the second stage assigns p_t using full resolution. In such a design, the problem of back-mapping arises: In a conventional one-stage design, the track parameters are output directly by the track finding. In the two-stage design, the track finding stage has to output pointers to the full-resolution data, such that the second stage can retrieve the full-resolution data from a pipeline memory and use them for assigning p_t. The conceptual simplicity of the one-stage template matching method, however, would be lost.

In view of the fine granularity of the detector and the high required p_t-resolution combined with the high number of detector channels, the non-projective chamber geometry and high bending power of the detector, the template matching method does not appear feasible within the available calculation time.

Histograming method

When employing a histogram method one has to find adequate histogram functions. In case of the track finder the best function values would be transverse momentum $p_t = f(\phi,\phi_b)$ and location $\phi_{track} = f(\phi,\phi_b)$ as a function of spatial and angular track segment coordinates ϕ,ϕ_b. In all stations but station three transverse momenta can be derived from the bending angle ϕ_b. In close vicinity to station three the bending angle ϕ_b has a zero crossing. Thus transverse momenta cannot be deduced from track segments in station three. Hence a method where the desired quantities, p_t and ϕ_{track}, are assigned by a function and detectable directly from the histogram can be ruled out.

An alternative possibility is to find functions of the hit coordinates (ϕ and ϕ_b) giving a calculated hit coordinate in a reference plane. Since such a function for station three cannot be found station three has to be the reference plane. Function values are spatial and angular coordinate of the tracks in station three. They are entered in the two dimensional ϕ-ϕ_b-histogram. A peak will form in the histogram when track segments (ϕ,ϕ_b) come from the same track and thus enter the same histogram bin. The location of the peak corresponds to the location of the track but not to the transverse momentum p_t. That means the histogram method can be used only to find tracks but not to assign a transverse momentum. In addition to the number of entries for each bin one has to store the relative address of the track segments which caused the entry. Once the peak in the histogram is found one can select the track segments of the perceived track(s) using their addresses stored with each bin. They are used to find the hit coordinates to calculate the transverse momentum p_t. However, the compactness of the histogram method is lost.

For the task of assembling track segments to a complete track the full ϕ-resolution (0.3 mrad or 11 bits) and ϕ_b-resolution (10 mrad or 8 bits) is not needed. Assuming that eight bits for ϕ and five bits for ϕ_b are sufficient, the histogram still has a dimension of size 256 (ϕ) times 32 (ϕ_b). Each histogram bin has to store the number of entries and the addresses of at least four track segments which caused the entries. This means the peak finder has to find the highest entry in a (256 times 32 =) 8192 bin histogram. Given the timing constraints this is also not practicable.

While proceeding in such a manner is a common approach in software solutions a hardware implementation does not seem practicable. The size of the histogram requires a huge amount of logic units. Moreover the calculation time would by far exceed the required maximum latency.

The strength of the histogram method, namely producing a histogram with bins of the desired features, can not be exploited to the full extent. No function can be found for all input data which produces the transverse momentum p_t and location η, ϕ. Therefore the architecture loses its compactness.

Neural networks

Recently intensive research has been conducted on the application of neural nets in high energy physics [17]. This includes software and off-line triggers as well as hardware triggers. In several more high energy physics experiments neural networks are considered for application and some are already in use (CDF [18], CP-LEAR [19], H1 [20,21], NEMO, WA92 [22]). However, it has to be said that until now no first level trigger was employed using neural nets only. This is due to the relatively long processing time and to the limited complexity of implementable algorithms. The response time of

typical commercially available digital neural networks is found to be between 1 to 10 ms. A recent application of a digitally programmable analogue neural network [23] is reported in [20]. The processing time in the described application is as low as 50 ns. Analogue designs, however, typically have precisions of a few percent. The position input to the track finder processor has a resolution of 11 bits, corresponding to a precision of 0.05%. Analogue designs are therefore out of the question.

Concluding the state of the art of hardware implementation of neural nets it must be said that today it is not sufficiently advanced. However, as designs of neural net implementations evolve, especially with respect to processing speed and number of inputs, they can become a possible solution for first level triggering and thus for the track finder.

Extrapolation method

A typical software approach, the extrapolation method was elaborated for the hardware implementation.

The basic principle is to attempt to match track segments caused by the same track. This is done by extrapolating into the next station from a track segment using the spatial and angular measurement.

While pattern matching methods usually deliver the wanted track property directly, the extrapolation method requires three steps:

- pairwise matching of track segments by extrapolation
- assembling track segment pairs to full tracks
- assigning the track properties transverse momentum and location.

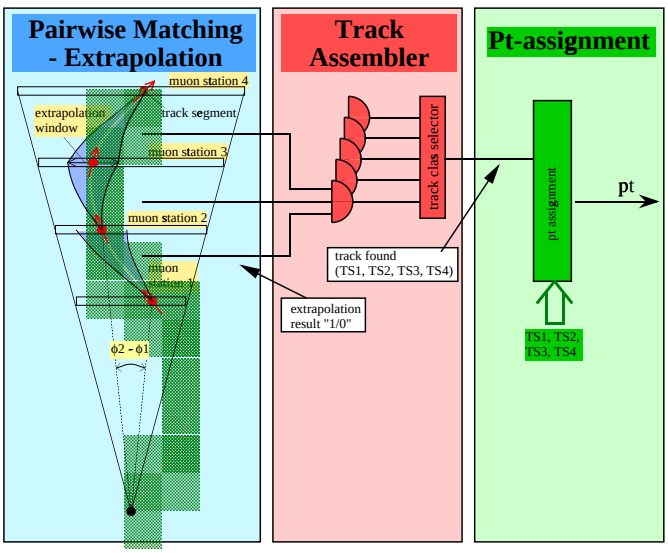

Fig. 5.: Principle of the track finder algorithm (3-Step scheme).

Fig. 5 illustrates the principle of the track finder algorithm. Global track finding methods such as the pattern match method directly set the input data into relationship to the wanted features. While this would be an advantage in many applications, in this case it complicates the implementation. When performing a pattern match one looks for track segment combinations which belong to one track (track finding). For this task full measurement precision is not needed. However, after a pattern has been recognized the link to the original track segment data is not available any longer. Thus patterns must be formed by track segment data with full resolution so that each pattern allows directly the determination of the wanted features with high precision.

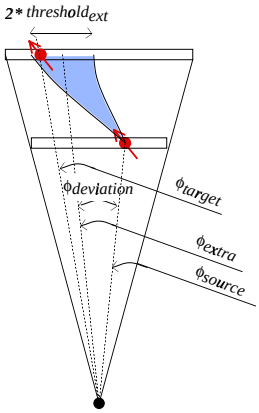

Fig. 6.: If a track segment is found to be within the extrapolation window given by ϕ_{extra} and $threshold_{ext}$, the extrapolation is considered successful.

However, the extrapolation method splits the process of measuring the transverse momentum p_t into three steps. This approach allows for a flexible use of the resolution in each step according to the requirements. The track finding is performed with reduced resolution, thus resulting in a reduced number of track patterns to recognize. As a consequence the hardware expense is smaller and the execution time shrinks. For assigning the transverse momentum p_t the full resolution of the track segment measurements is still available.

EXTRAPOLATION METHOD

The pairwise matching is based on the principle of extrapolation. Using the spatial coordinates ϕ_{source} and the angular measurement $\phi_{b,source}$ of the source track segment an extrapolated hit coordinate ϕ_{extra} in another chamber may be calculated. If a target track segment is found to be at the extrapolated coordinate within a certain extrapolation threshold $threshold_{ext}$ the match is considered successful (see fig. 6).

$$\phi_{extra} = \phi_{source} + \phi_{deviation}(\phi_{b,\,source}) \qquad \text{(Eqn. 1.)}$$

$$\left|\phi_{extra} - \phi_{target}\right| \le threshold_{ext} \qquad \text{(Eqn. 2.)}$$

Extrapolation feasibility

Simulations have been conducted to prove the feasibility of the extrapolation between the stations [3,4]. Fig. 7 shows the relation between the bend angle ϕ_b in the source station and the deviation of the particle track between target- and source-station ϕ_{target} - ϕ_{source} for several station pairs. The graphs show unambiguous relationships proving the feasibility of extrapolation between these station pairs. The same condition can be found in all other station pairs except for those extrapolating from station three. Fig. 8 shows the situation

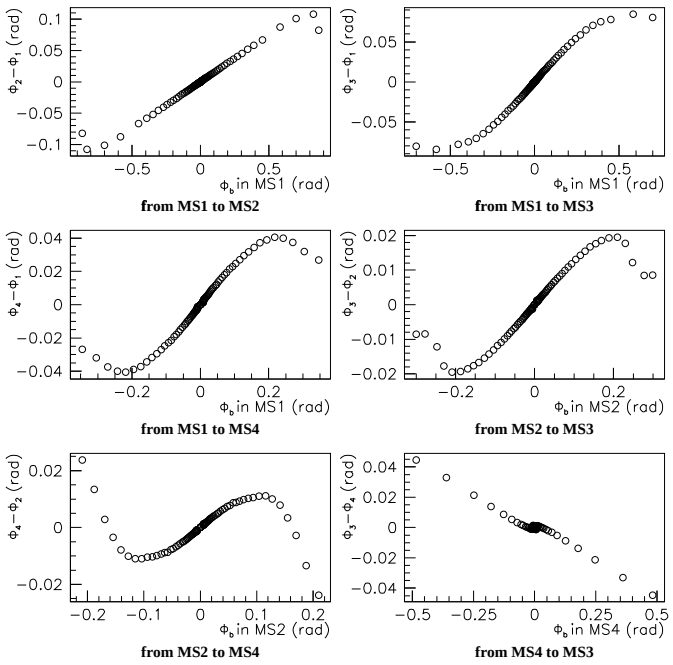

Fig. 7.: Relationship between bend angle measurement ϕ_b in the source station and deflection of a muon ϕ_{target}-ϕ_{source}.

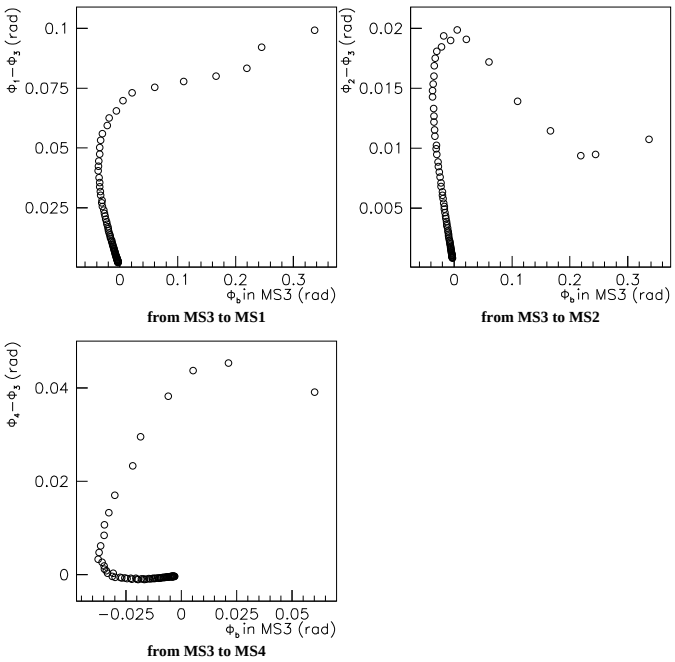

Fig. 8.: Relationship between bend angle ϕ_b in station three and deflection of the muon is ambiguous.

when extrapolation is done from station three. No unambiguous relationship can be found. Moreover, for small bend angles no prediction can be done at all. This effect is caused by the zero crossing of the bend angle. However, since all other extrapolations are feasible these problems can be circumvented by extrapolating towards station three instead off station three (see left part of fig. 5).

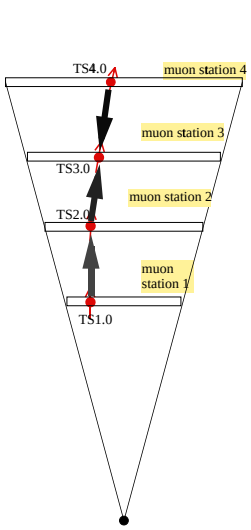

Fig. 9.a: Matching track segment pairs are combined to a full track.

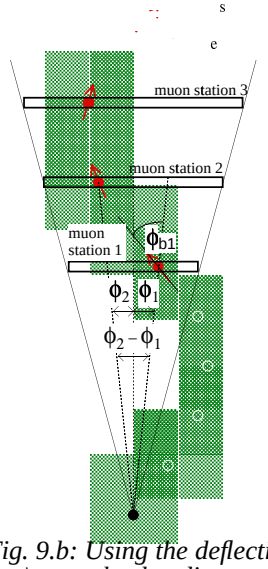

Fig. 9.b: Using the deflection ϕ_2-ϕ_1 or the bending angle ϕ_b to determine p_t.

Track segment assembling - Acceptance study

Once track segment pairs are found they are linked together to full tracks. Track segment pairs of different chamber pairs may be matched to each other if they have one track segment in common. This scheme is illustrated in fig. 9.a.

However, due to geometrical inefficiencies and chamber failures missing hits must be accounted for. Simulation studies for the acceptance of muons were conducted for two possibilities [3]. The first option is to require tracks consisting of at least three out of four track segments or track segments in the two innermost stations. The second option is to require tracks with at least two out of four track segments. Fig. 10 compares the results. When simulating the requirement of three out of four matching track segments only about 82% of all muon tracks are found. An acceptance of more than 95% is achieved when the requirements are loosened to two out of four matched track segments. Consequently the track finder accepts tracks consisting of only two matching track segments. That means even a single track segment pair is already considered a valid track.

P_t - assignment

In order to assign transverse momentum p_t we use the track's bend angle. Two methods are available [3,5,6,24]. One method uses the difference of positions in two distinct stations (see fig. 9.b). Two spatial coordinates are sufficient. Fig. 11 illustrates the relation between transverse momentum and aforementioned difference. One expects the absolute value of the difference in bend angle to decrease with increasing p_t. However, due to the zero crossing of the bend angle in station three the absolute value of the difference of the angles at first rises with p_t (except for ϕ_2-ϕ_1). An unambiguous relationship between difference of positions and transverse momentum p_t is shown for difference ϕ_2-ϕ_1. For other station pairs this relationship is ambiguous. In such a case the bend angle

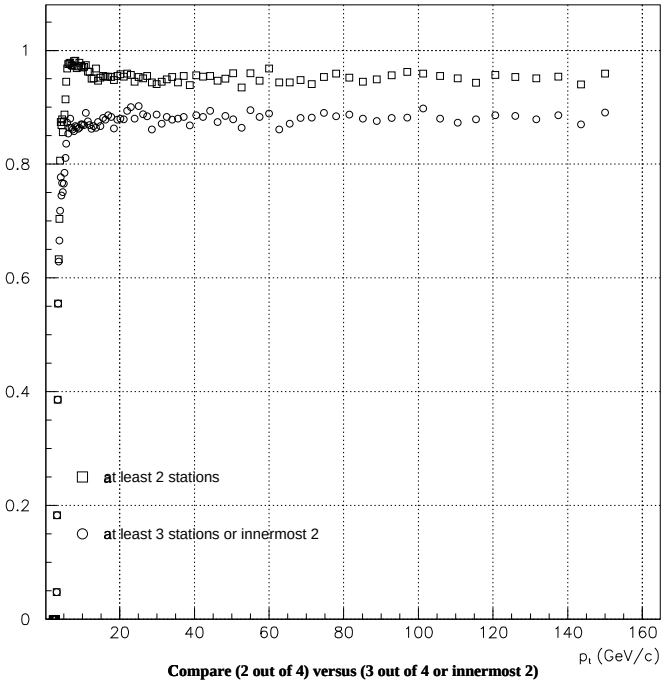

Fig. 10.: This plot compares two track finder requirements. Circles show the case where a track has to have at least three track segments or track segments in the two innermost stations. Squares require only two track segments for a valid track.

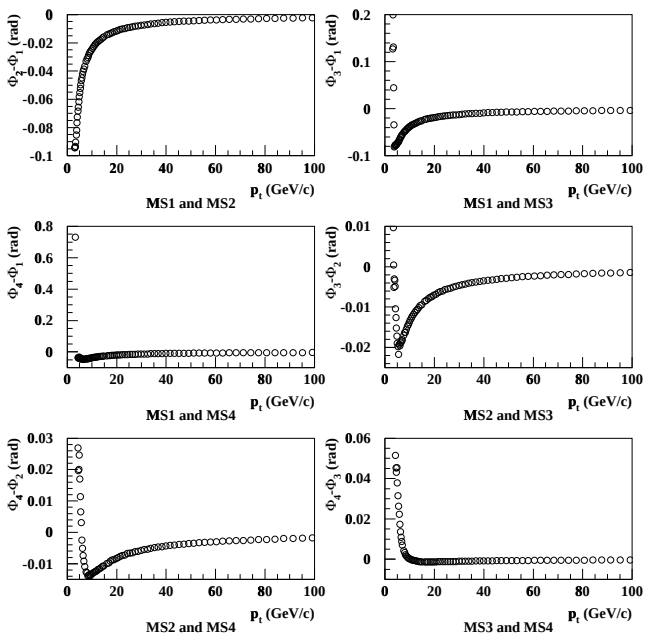

Fig. 11.: Difference of azimuthal hit coordinates ϕ_{ii}-ϕ_i (deflection of the muon between two stations) over transverse momentum p_t for several station pairs.

measurement of a single track segment can be used to measure p_t [3,4].

Figure 12 shows the p_t-resolution achieved by the method described.

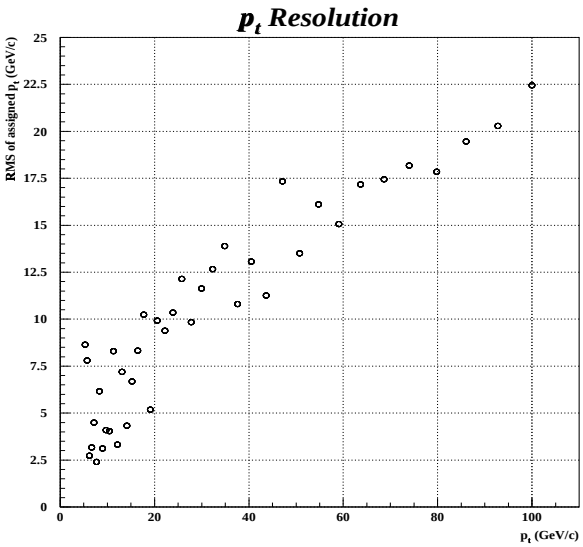

Fig. 12.: Resolution of the transverse momentum p_t as a function of this momentum itself.

The drift tube trigger primitive generator also delivers a quality information. This quality indicates how many layers of a chamber contributed to a track segment. It is also an indication of the measurement resolution of the track segments [2]. These quality bits are used to select the algorithm in order to provide the most accurate p_t-measurement.

Performance

The presented simulation results show that extrapolation, track assembling and p_t assignment are possible. Fig. 13 shows the efficiency of the track finder processor for p_t-thresholds of 20, 40, and 50 GeV/c [3]. The results take background into account by superimposing on average 20 minimum-bias events and by using a full simulation of the particle interactions with the material of the detector. The momentum resolution is given by the steepness of the efficiency curves at the nominal threshold values.

ARCHITECTURE AND HARDWARE ALGORITHM

The basic architecture of the track finder processor [16] is described. The mapping of the chamber structure onto the hardware level is discussed. 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 later described FPGA prototype was designed [4,5,6,16,25,26].

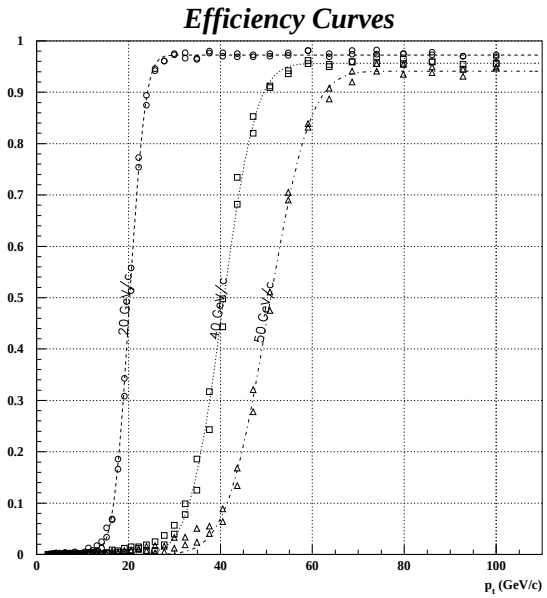

Fig. 13.: Efficiency curves for p_t-thresholds of 20, 40 and 50 GeV/c. Higher thresholds are not likely to be needed.

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 (480 track segments times 22 bit per track segment) data per crossing cycle yields an input data rate of about 400 Gb/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 [16].

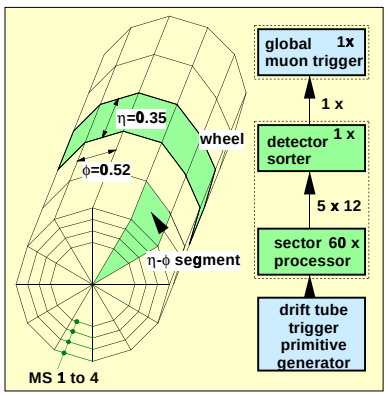

Fig. 14.: Logical segmentation of the track finder processor.

In order to render communication between processing units possible at a minimum expense, the logical structure of the chamber system is mirrored inside the track finder processor hardware. In fig. 14 the logical segmentation of the track finder processor is shown.

A sector processor matches the track segments identified by the drift tube trigger primitive generator logic [2] and tries to form up to two complete tracks. If the sector processor succeeds, it assigns a transverse momentum p_t, and determines the location in ϕ and η of each track. Tracks which traverse more than one detector segment are given out by the sector processor of the detector segment where the track's innermost track segment is found.

Of the 60 (12 ϕ-sector 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 detector 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 [1, 27].

TRACK FINDER PROCESSOR ALGORITHM

In fig. 15 a 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) [16].

The extrapolation unit EU attempts to match track segment pairs of distinct stations using the extrapolation criteria described earlier. When track segment pairs meet these criteria the information is forwarded to the track assembler TA.

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).

Short description of the track finder algorithm

In the following, a short overview of the track finder algorithm is given (see fig. 15). The algorithm reduces the number of possible track candidates from about 1800 to two. Fig. 16 illustrates the first part of the reduction.

The extrapolators (EXT) match track segment pairs to each other. For each possible track segment pair the information bit *er* indicating whether the 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. 16 b, c). Recognizing 1800 patterns or track candidates can be done easily by employing a pattern

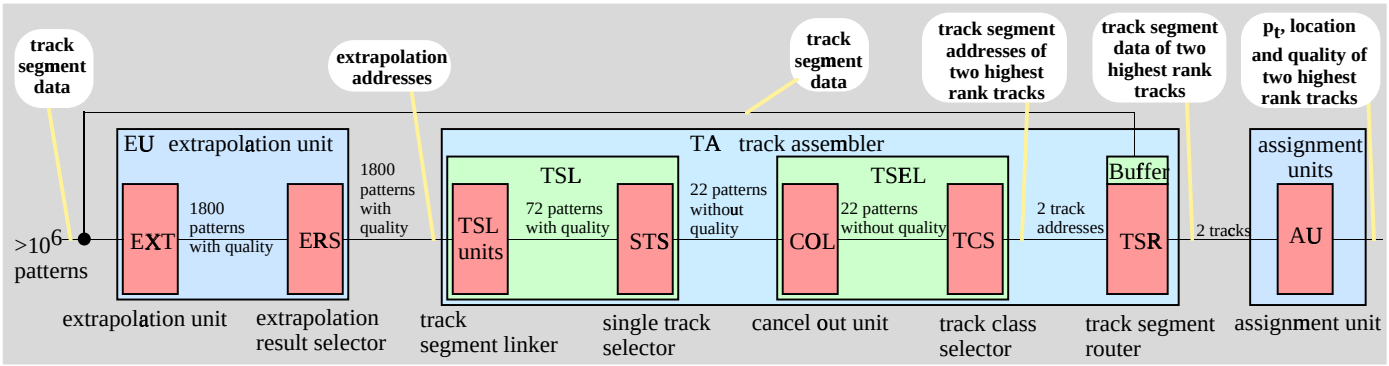

Fig. 15.: Block diagram of a sector processor.

comparison method. However, each of the patterns has a quality information attached to it. In order to select the two highest rank patterns a comparison of quality numbers would be necessary. This takes too much time.

The extrapolation result selector (ERS) selects the two best extrapolations for each source track segment. It encodes the extrapolation result bits *er* into address words *adr* each indicating the target track segments of up to two successful extrapolations. The extrapolation quality *eq* is given out (fig. 16 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. 16 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. 16 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 equal 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. Selection criterion is the number of track segments involved in a track candidate.

The track segment router (TSR) uses these relative addresses mentioned above to extract the track segments data out of the buffer memory and outputs the corresponding track segment data.

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

HARDWARE IMPLEMENTATION

Extrapolation

Extrapolation between six possible station pairings are conducted in parallel (1-2, 1-3, 1-4, 2-3, 2-4, 4-3). Muons can cross detector segment boundaries. Thus it is necessary also to compare track segments with track segments from at least five neighbouring detector segments (two adjacent sectors in ϕ and three adjacent segments in η). Consequently for each possible start track segment (two in each chamber) twelve target track segments have to checked whether they fulfil the extrapolation

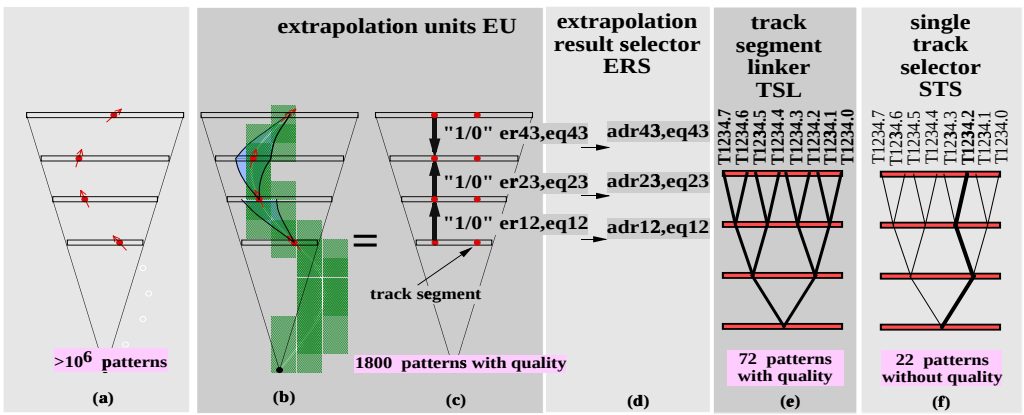

Fig. 16.: Reduction of possible track candidates by the track finder algorithm.

criteria. Thus in total 144 (six station pairings times two track segments per chamber times twelve target track segments) have to be conducted. Fig. 17 shows the block diagram of an extrapolation unit responsible for an extrapolation from one source track segment to twelve target track segments. The output are the extrapolation results *er* and the extrapolation quality word *eq* for each of the target track segments. For calculation of the extrapolation value static memory based lookup tables are employed. The comparison is done using subtractors and window comparators. The extrapolation result selector (ERS) employs priority encoders.

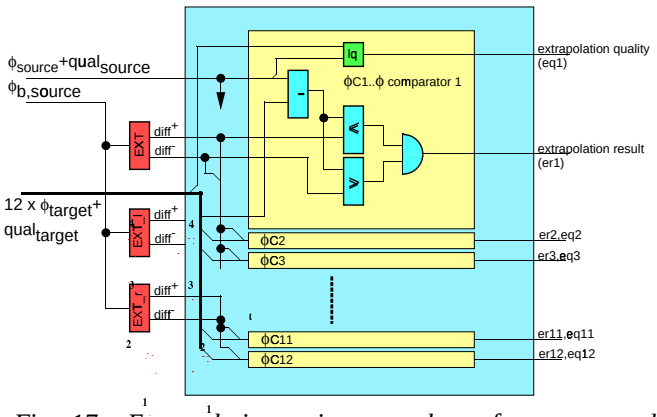

Fig. 17.: Extrapolation unit extrapolates from one track segment and compares the extrapolated value to twelve target track segments. It outputs the extrapolation result (er) and the extrapolation quality (eq) for each comparison.

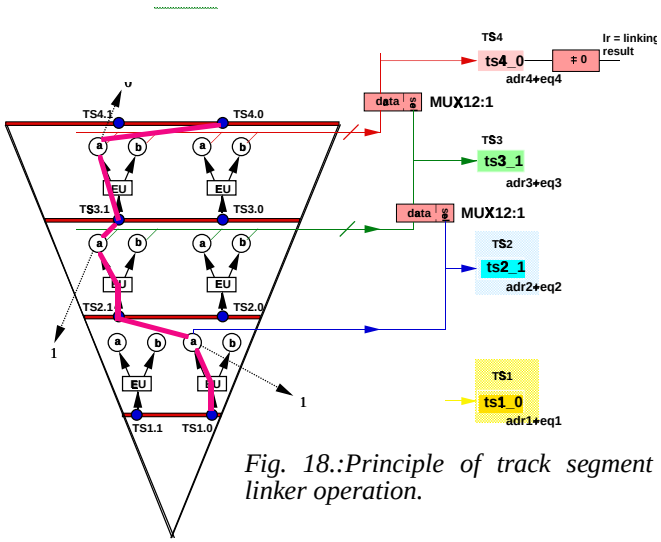

Fig. 18.:Principle of track segment linker operation.

Track assembling

The extrapolation result selectors (ERS) output the addresses of target track segments of successful extrapolations. The track assembler employs a dynamic track segment linking scheme [16] to assemble track segment pairs to up to two full tracks. An example is illustrated in fig. 18. In the example it is assumed a muon track is composed of four track segments (Track segment 0 in station one and four and track segment 1 in station two and three). Using the addresses provided by the extrapolation result selectors (in the extrapolators) the scheme assembles the track in a sequential way. The extrapolator EU extrapolating from station one to two outputs the address for the target track segment TS2_1. The extrapolator starting the extrapolation from this track segment outputs the target track segment TS3_1 and so on. The hardware implementation employs simple multiplexers (fig. 18). Target track segment addresses of extrapolation between station one and two are routed to the select input of the multiplexer. Target track segments of all extrapolators of extrapolations between stations two and three are routed to the data input. As a consequence the multiplexer outputs the target track segment address in station three. This architecture is repeated for extrapolations between station three and four. Once the addresses of matching track segments are known, the track segment data are extracted from a buffer memory using multiplexer arrays (TSR).

Assignment

The transverse momentum p_t is assigned using static memory based lookup tables. Location ϕ and η can be derived directly from the measured track segment data and track segment addresses respectively.

FPGA PROTOTYPE

This section describes the prototype of the muon track finder processor.

The goals of the realisation of the FPGA-prototype were:

- to demonstrate that the VHDL-model of the processor can be implemented in hardware with reasonable expense;
- to show that the designed and simulated algorithm also works implemented in hardware and
- to show that the general design concept is feasible.

However, for economical reasons the XILINX 4000 technology was employed (and not an ASIC). It was not our aim to build a prototype capable of fulfilling timing specifications of the CMS first level trigger [8].

FPGAs with an I/O pin count of not more than 192 pins were employed. Considering this restriction we used each I/O pin to insert or extract two data bits to or from each physical unit within one clock cycle. As the used technology does not allow a synchronously working design with a clock frequency in excess of 50 MHz the internal clock frequency was designed to be 20 MHz. As a consequence the I/O clock frequency yields 40 MHz. It is obvious that this is no option for a final implementation. However, as mentioned earlier, the FPGA-prototype was not designed to fulfil timing specifications of CMS.

In all, the FPGA processor employs 19 FPGAs and 19 lookup tables. A total of 240000 FGPA gates (only 60 to 70% are used) or 10000 cellular logic blocks are available. 10000 component pins are on the printed circuit board. 1236 input bits are brought onto the board each clock cycle and 362 output bits are given out each cycle. As the board is operated in time multiplexed mode only half the number of I/O pins is necessary, i.e. 618 input pins and 181 output pins. The FPGA-processor

evaluates each event within 29 cycles (14 cycles are required for the final system). The printed circuit board is designed in 9U VME standard. However, due to the large number of I/O bits the board is about 15 cm longer than the standard VME board.

As the FPGA-prototype demonstrates, the logic implementation of the designed algorithms does not pose a problem for implementation. The number of necessary logic gates is comparably low. However, the I/O count of the physical units is very high. In the prototype design only a reduced number of bits is processed. For the final implementation a bit number of some 2500 has to be processed each cycle in each sector processor. This task could become even more daunting because discussions are ongoing to increase the amount of data provided by the trigger primitive generators.

There are several options to come by the problem. One is transmitting a group of track segments sequentially with 80 MHz. Another is transmitting in a bit serial format using an optical link receiver or fast serial copper link directly on the board. However, this problem is not solved yet.

The design of the prototype board showed clearly that today's FPGAs are not suited to the track finder requirements. Both the I/O count of the packages and the data propagation do not allow the final design of the track finder system employing today's FPGA.

However, the functionality of the implementation of the track finder algorithm, the momentum measurement algorithm and the hardware structure mapping of the detector geometry onto the hardware level was proven to be adequate to the system requirements. Moreover it could be shown clearly that the VHDL-model can be implemented in hardware with reasonable hardware expense.

A feasibility study for a possible final implementation employing application specific integrated circuits (ASICs) has been conducted [16,26]. It clearly demonstrated that today's ASIC technology is sufficiently advanced to implement the processor fulfilling all requirements.

CONCLUSION AND FURTHER PERSPECTIVES

In the chapter 'implementation options' and also in [16] it is shown that no previously implemented system is suitable to be applied in the track finder processor environment. Conventional methods applied in hardware triggers fail for the track finder. Especially the most common approach, the pattern comparison, must be ruled out because of the large hardware extent. Instead the extrapolation method is introduced. It is shown that the algorithm copes with the track finder specifications. The algorithm can be implemented with a minimum of hardware. Using VHDL and FORTRAN simulation the algorithm and its hardware representation were optimized. Simulation shows clearly that the simplicity of the design concept, namely reducing data flow in subsequent steps (by extrapolation, track assembly and property assignment) and selecting the highest ranking track candidate after each reduction step without sacrificing measurement accuracy, proves to be an efficient method. The FPGA prototype demonstrated that the algorithm (described in VHDL) can be implemented in hardware with reasonable effort. The prototype clearly shows the proper functionality of the implemented system. Using simple logic modules, such as multiplexers, comparators, subtractors and logic gates, proves to be an important key point for the success of the design. However, it is pointed out that the number of bits to be processed in parallel poses a challenge to the hardware implementation.

The design of the track finder processor, as it is introduced in this work, cannot at all be regarded as terminated. Although both the simulation and the prototype already delivered satisfactory results the track finder design presented here represents only a first step towards final implementation. The work suggests an algorithm and an implementation method. However, as the surrounding environment of the processor will evolve, more and more both the specifications and the implementation of the track finder processor will have to be refined accordingly.

REFERENCES

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

[2] 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

[3] T. Wildschek, Design and Simulation of the CMS First Level Muon Trigger Track Finder, Dissertation, Technische Universität Wien, 1998.

[4] 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.

[5] 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.

[6] 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.

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

[8] A. Kluge, W. Smith, CMS Level 1 Trigger Latency, CMS TN/96-33, CERN, 8 March 1996

[9] Henk W. den Bok et al., Track recognition with an associative pattern memory, Nucl. Instr. and Meth. A300 (1991) 107-114.

[10] M. Dell'Orso, L. Ristori, VLSI Structures for Track Finding, Nucl. Instr. and Meth. A278 (1989) 436-440.

[11] S.R. Amendolia et al., The AMchip: a VLSI associative memory for track finding, Nucl. Instr. and Meth. A315 (1992) 446-448.

[12] S.R. Amendolia et al., The AMchip: a full-custom CMOS VLSI associative memory for pattern recognition, IEEE Transactions on Nuclear Science, Vol. 39, No.4, 1992, 795.

[13] T. Kohonen, Content Addressable Memories, 2nd Ed., Springer-Verlag, 1987.

[14] Cypress Semiconductor, Data Sheet Advanced Information Cy7C915 1k x 42 SmartCAM.

[15] Music Semiconductors, Data Sheet MU9C1640 CacheCAM, June4 1993.

[16] A. Kluge, The Hardware Track Finder Processor in CMS at CERN, Dissertation at the Technical University of Vienna, October 1997, CERN-THESIS-98-016

[17] H. Kolanski, Application of Artificial Neural Networks in Particle Physics, DESY 95-061, April 1995, ISSN 0418-9833 or Nucl. Instr. and Meth. A367 (1995), 14-20.

[18] C. Loomis and J. Conway, "Using an Analog Neural Network to Trigger on Tau Leptons at CDF", Proceedings of AIHENP95, 1995.

[19] G. Athanasiu, P. Pavlopoulos and S. Vlachos, "A neural network trigger system for the CP-LEAR experiment", Proceedings of AIHENP95.

[20] S. Schiek, G. Schmidt, Application of a high speed analog neural network chip for first level triggering at the H1-Experiment at HERA, Proc. of the "International Conference on Artificial Neural Networks" ICANN'95, Oct. 9-3, 1995, Paris, France, Vol. 2, pp.363-368, ISBN 2-910085-18-X.

[21] Fent et al., The Realization of a Second Level Neural Network Trigger for the H1 Experiment at HERA, AIHENP'96, Lausanne, Switzerland, Sept.2-8, 1996.

[22] C. Baldanza, Results from a neural Trigger Based on the MA16 Microprocessor, Int. J. Mod. Phys. C6 (1995)567 or DFUB95/2, 1995.

[23] K. Hoen et al., 70 input 20 nanosecond pattern classifier, The 1994 IEEE International Conference on Neural Networks, Vol. 3, 1854-1859.

[24] T. Wildschek, Simulation of the Silicon Tracker/Vertex Detector of the ATLAS Experiment at the Large Hadron Collider at CERN, Diploma Thesis, Technische Universität Wien, 1993.

[25] A. Kluge, T. Wildschek, The Hardware Muon Track Finder Processor in CMS - System and Algorithm, CMS Note 1997/092

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

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