6 Electron trigger with the TRD

6.1 Physics motivation

The study of rare probes with cross sections in the order of ≈ 100 μb (corresponding to probabilities per collision of $\approx 10^{-5}$) to diagnose the properties of the QGP requires dedicated triggers to enhance the events containing the signals. Especially electromagnetic probes that are not affected by hadronic reinteractions and therefore provide the most direct view of the reaction scenario need a substantial effort due to their small cross section. Equally rare are the electromagnetic decays of hadron resonances, of which the heavy vector mesons J/ψ and Υ are of special importance due to the expected signatures of their yield for specific plasma conditions [1–3].

The ALICE data acquisition has to serve all the different parallel trigger requests for the various physics observables; for the sake of the present discussion it is assumed that the bandwidth available for high p_t electron physics is limited to an equivalent of 20 Hz of central events. Under those conditions about $2 \cdot 10^7$ events per year of ALICE operation will be recorded for the TRD trigger.

To test various models and resolve ambiguities it is of utmost importance to measure differential distributions, i.e. the transverse momentum (p_t) and the centrality dependence of the signals. A trigger is especially needed for i) large transverse momenta of the resonances that are typically suppressed due to the exponential fall-off of the spectra and ii) for large impact parameter events for which the probability to produce the interesting probes is substantially reduced.

In order to make a variety of physics signatures accessible to ALICE with sufficient statistics, the TRD trigger is designed to i) find and select tracks with transverse momenta of **more than 3 GeV/c**, ii) separate **electrons** from pions and iii) allow to compute correlation quantities like the invariant mass of track pairs or the multiplicity in spatial regions.

The physics observables that benefit from the TRD trigger are:

- J/ψ production at large transverse momentum
- Υ production
- The Thermal Dilepton Continuum in the invariant mass range from 4 to 9 GeV/c^2
- Jet production with Jet Energies of more than 100 GeV

The di-electron capability of the TRD will be used to measure observables at midrapidity and thus delivers (complementary to the di-muon arm) valuable information for the diagnostics of the plasma under the most clean conditions, i.e. at the highest energy density combined with the lowest net baryon density. An unique advantage is the possibility to correlate rare probes with the other information obtained with the central ALICE detector in the same rapidity interval on an event-by-event basis.

To make efficient use of the solid angle available for the TRD, the trigger system has to cover the whole range in rapidity and transverse momentum with a high and uniform efficiency. The system is designed for a minimum bias measurement since important information is contained in the production rates the rare probes for different impact parameter [2, 4].

There is a large uncertainty about the multiplicity to be expected for Pb-Pb collisions at LHC energies (see discussion in Chapter 12). The system works well even for the maximum possible central multiplicities of $dN/dy = 8000$, but substantial improvements are achieved for minimum bias conditions or if the central multiplicity is significantly below 8000.

6.2 Concept

The TRD trigger (and the whole detection scheme) is implemented to select on high p_t electron pairs exploiting the transition radiation signature on a 5 μs time scale. Therefore the trigger scheme is organized in the following way (see Fig. 6.1):

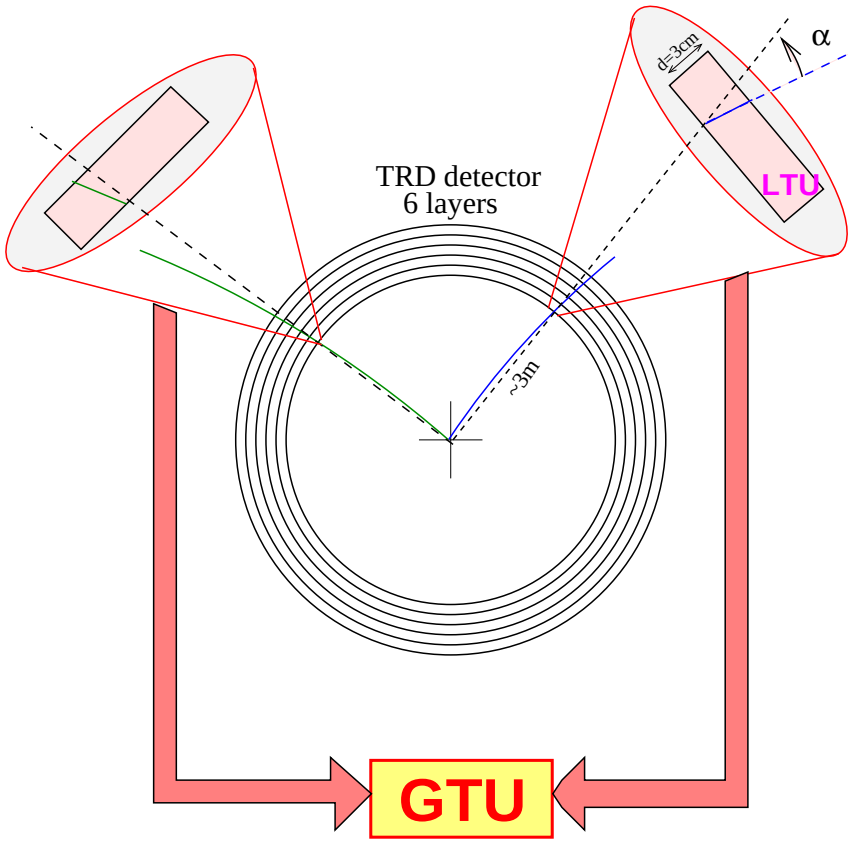

Figure 6.1: Trigger scheme.

1. Local track segment (*tracklet*) search independently in all chambers of the detector in parallel processors called Local Tracking Units (LTU).
2. Selection of stiff *tracklets* by means of maximal deflection compared to that expected for straight ($p_t = \infty$) trajectories.
3. Computation of particle identification (PID) information based on the total energy loss and the depth profile of the deposited energy.
4. Shipping of data from all stiff tracklet candidates ($p_t \geq$2 GeV/c) to a global tracking unit (GTU).
5. Selection of high transverse momentum candidates by requiring a sufficient number of merged tracklets (3 out of 6) with a global deflection corresponding to $p_t \geq$2.7 GeV/c.
6. Computation of the global particle identification information from combination of the local PID measures.
7. Global counting of positive and negative track candidates (possibly also within a given area).
8. Computation of two-particle correlation quantities like the invariant mass.

9. Transmission of the result to the ALICE trigger (Central Trigger Processor).

The identification of interesting physics signals by the TRD requires to inspect all Pb-Pb interactions for the occurance of rare probes, i.e. the trigger has to run at minimum bias event rates. Since the gating grid of the ALICE TPC cannot be opened at this rate the trigger decision based on the TRD detector alone has to become available on a time scale that is short as compared to the total drift time of the TPC ($T_{TPC} = 88$ μs). A decision time of about 6 μs is considered acceptable since the delay in opening the gating grid involves at most a shortening of some tracks in the TPC at large forward/backward polar angles. Alternate scenarios like opening the gating grid with the minimum bias trigger rate and closing it with the non occurance of the TRD trigger are also under discussion. In any case the whole trigger sequence needs to be completed within a maximum decision time of about 6 μs.

With the envisioned maximum multiplicities of up to 20000 primary charged particles in the acceptance of the TRD detector the implementation of the trigger scheme dictates a massively parallel computing model. The most demanding part is the online tracking of the full event with high enough quality to select events that occur with probabilities of the order of 10^{-5}. The whole architecture described in the following is optimized to achieve this goal.

6.3 Hardware implementation

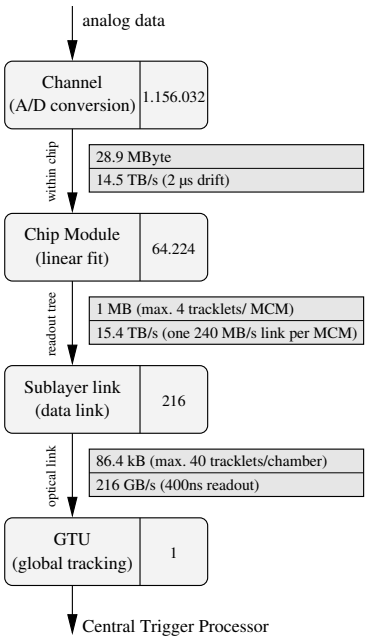

Figure 6.2: Trigger System Overview

Within the detector enormous amounts of data are produced on the lowest level, which have to be processed in a very short time. To reduce the incoming data volume as early as possible successive selection steps are implemented. Figure 6.2 shows the different processing steps with the associated amount of data.

6.3.1 Trigger concept

The basic idea of the trigger system is to find high-momentum electrons and separate them from pions by a reconstructed track line and a transition radiation (TR) signature. The electron-pion separation is performed via the TR photons, which are primarily detected at the end of the drift-time (see Figs. 11.10 and 14.20). The track reconstruction algorithm takes the known track model for high-p_{t} particles into account. Such particle tracks are essentially perpendicular to the readout chambers pad plane, neglecting the Lorentz angle and the fact that the chambers are flat. For a detailed geometric layout of the chambers, refer to Chapters 2 and 4. Only a small number of channels is required to read out a complete track, allowing for the implementation of a *tracklet* reconstruction engine in a highly parallel fashion. In the

following we will refer to a *tracklet* as the segment of a track in one readout chamber. Tracks with a large inclination angle, and thus a low transverse momentum, are not included in the tracking model and are therefore ignored. All space points above one given pad are considered to belong to the same *tracklet* and are included in a straight-line fit. This method performs worse as the occupancy increases and eventually fails in case of a multi-track pileup above a given pad. This has been taken into account in the microscopic simulations of the trigger (see Section 6.4). Tracks with a large deflection angle crossing a stiff track will distort the position resolution in the region of overlap. However, the corresponding pileup clusters can be detected at the hit level and will be excluded from the fit.

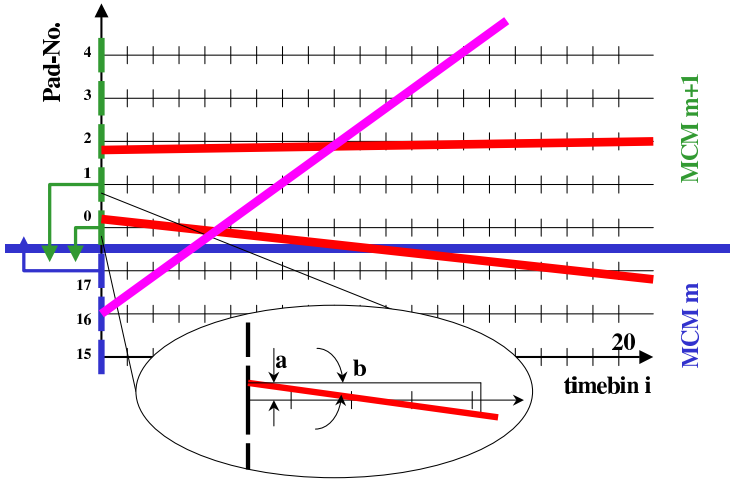

Figure 6.3: The *tracklet* fitting principle. The pad row runs in y direction. The drift direction is radial.

An overview of the tracking principle is sketched in Fig. 6.3. The ordinate identifies the pad row of the detector, along which the track is bent in the magnetic field. The abscissa shows the drift direction, discretized in a number of time bins (configurable up to 32). The trigger system calculates for each time bin and channel the position of a cluster based on charge sharing.

Due to the pad response function, on average, 2.4 pads are hit by a high momentum track. Pads with a local maximum in charge deposit are selected and the precise y position of the corresponding hit is determined from the pulse height of this pad and its two neighbors for every time bin. The position is determined based on lookup tables and indexed by the pad amplitude ratios center/right and center/left. For a well defined charge sharing one of this ratios suffices for position reconstruction. Given the involvement of three pads and therefore a second redundant position measurement per time bin, pileup hits can be detected and rejected on a time bin basis. Refer to Section 6.4 for a detailed discussion of the online chamber position resolution.

The trigger is designed to perform a straight-line fit based on the calculated space points in y as a function of drift-time. As described in more detail in the next section, during the drift-time, for each channel the input parameters for a straight-line fit are calculated and updated in sum memories. After the drift-time, the trigger processor calculates the final *tracklet* parameters such as the slope, intercept, variance, etc.. Only *tracklets* are considered, which involve a maximum of four pads, corresponding to a maximum y-deflection of one pad. After this operation each individual *tracklet* candidate is subject to an angle and (possibly) particle identification (PID) cut, selecting high-p_t *tracklets* . The PID is based on energy deposit (see Sections 6.4.5 and 11.5). Low momentum particles have a large deflection in the bending plane. They will be rejected either by slope or they will not meet the criterion on minimum number of space points required for a valid *tracklet* . The *tracklets* identified as high p_t (electron) candidates are shipped through a read out tree to the global track matching unit, which combines the *tracklets* of the six TRD layers into one track with improved momentum resolution and PID.

The whole of TRD electronics is arranged in groups of 18 channels (adjacent pads in the same pad row), mounted on multi-chip-modules (MCM). The charge-sharing and finite deflection of high-p_t

tracklets require the neighboring MCMs to share the data of their borderline channels. For example, for the position determination via charge-sharing of channel 17 (counting from 0) of MCM m, channel 0 of MCM $m+1$ is required and vice versa. In order to not have tracking inefficiencies at MCM borders, *tracklets* with the maximum foreseen inclination have to be considered, as sketched in Fig. 6.3. One possible implementation could be the exchange of the content of the *tracklet* SUM memories, which is calculated during the drift-time, at the end of the drift-time for the calculation of the complete *tracklet* parameters. However, this would result in an increased processing latency due to the required data exchange, which cannot be pipelined. On the other hand, the inclusion of a second channel (for example channel 1 of MCM $m+1$) would allow the complete reconstruction of a boundary-crossing *tracklet* on MCM m. Note that no further efficiency will be gained by the shipping of channel 16 of MCM m to MCM $m+1$ as the given *tracklet* will already be reconstructed by MCM m. Therefore, the total number of inputs on each 18-channel MCM is 21, where two channels are shipped to and one is received from MCM $m-1$ and one channel is shipped to and two are received from MCM $m+1$.

Simulations have shown that communication between two chips in z direction is not necessary (see Section 6.4). *Tracklets* split in that direction can be recombined at the level of the global tracking unit.

6.3.2 Local Tracking Unit (LTU) functionality

An overall picture of the electronics is given in Fig. 5.2. The LTU functionality includes everything after the ADC. It comprises a so called *tracklet* preprocessor (TPP), which includes storage of the raw ADC data in the event buffer, a MIMD microprocessor, which subsequently computes and selects the *tracklet* and the read out part.

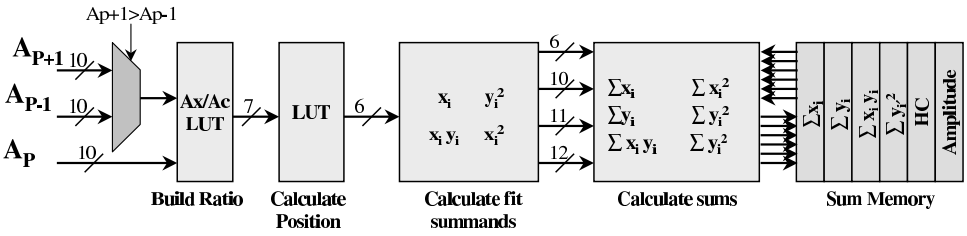

Figure 6.4: Functionality one dedicated channel of the preprocessor.

The TPP performs data acquisition from the 21 ADCs and, in parallel, executes the required *tracklet* preprocessing in order to speed up the determination of the track parameters after the drift-time ends. Its functionality is shown in Fig. 6.4. The tracking model assumes a straight-line fit according to:

$$y_i = a + bx_i, \tag{6.1}$$

where i is the time bin number. The resulting slope and intercept for N space points (x_i, y_i) are defined as:

$$b = \frac{N\sum x_i y_i - \sum x_i \sum y_i}{N\sum x_i^2 - (\sum x_i)^2} \qquad , a = \frac{\sum x_i^2 \sum y_i - \sum x_i \sum x_i y_i}{N\sum x_i^2 - (\sum x_i)^2}. \tag{6.2}$$

The space points are given by the drift distance x_i and the measured position y_i along the wire. Although the y-position resolution depends on the total charge of the cluster, the fit is not performed with weights as those would have required more hardware. All N clusters are just required to have a minimum total charge (the sum of three neighboring pads), which can be configured.

All digitized ADC values are stored in the event buffer. The various sums shown in equation 6.2 can be accumulated during the drift-time in a multi-ported register file (FIT), implementing read-modify-write cycles. At the end of the drift-time, all required input parameters for the fit are stored in the

appropriate registers of the fit register file. At that point, the fast clocks are enabled, starting the multiple–instruction–multiple–data (MIMD) processors and computing the *tracklet* parameters according to equation 6.2. This final computation requires only multiplication, addition and one division. The start and end time of the fit algorithm can be configured within an interval of up to 32 time bins.

A space point is always assigned to the pad carrying the maximum signal. However, valid *tracklets* can span two pads, which results in those *tracklets* being split over these two pads as their cluster maximum moves from one pad to the next (Fig. 6.3).

In order to avoid tracking inefficiencies, those *tracklet* segments have to be merged. The merging makes use of the constant width Δy of the individual pads which allows for the derivation of the combined sums from the two adjacent channels $(p, p+1)$. Parameters for the full *tracklet* with respect to channel p are computed by merging its data with the data from channel $p+1$:

$$\sum x_i = \sum_p x_i + \sum_{p+1} x_i \qquad , \sum x_i^2 = \sum_p x_i^2 + \sum_{p+1} x_i^2 \tag{6.3}$$

,

$$\sum y_i = \sum_p y_i + \sum_{p+1} y_i + N_{p+1}\Delta y \qquad , \sum x_i y_i = \sum_p x_i y_i + \sum_{p+1} x_i y_i + \Delta y \sum_{p+1} x_i, \tag{6.4}$$

$$\sum y_i^2 = \sum_p y_i^2 + \sum_{p+1} y_i^2 + 2\Delta y \sum_{p+1} y_i^2 + N_{p+1}\Delta y^2. \tag{6.5}$$

The sum of y_i^2 is required for the computation of a fit quality parameter.

This functionality is implemented by the MIMD processor. In order for a *tracklet* segment to be considered, it has a programmable minimum length of typically four measured space points. The equations are symmetric whether a left or right neighbor is merged. Therefore, it is sufficient to always have one predetermined side here, which will be the ascending pad number.

6.3.2.1 Tracklet Preprocessor

The TPP calculates parameters from stiff tracks of 18 data channels for a linear fit during the drift-time of 2 μs . All TPP configuration parameters can be set externally. The whole architecture (as shown in Fig. 6.5) can be split into three sections.

- The first part is the front-end interfacing to the ADCs, which are also implemented on the chip. This stage integrates the event buffers to store the raw data for later read out upon a L2 accept and a logic block to select space points in real-time for all channels simultaneously. Also, two lookup tables (LUT) per channel are implemented. The first LUT builds the ratios between the center and neighbor pads, and the second calculates the position on the current pad.

- The second part of the TPP calculates the parameters needed for the fit.

- The third part is the read-modify-write block (FIT), which allows for updating of the various sums in one clock cycle and provides support for the interface to the MIMD processor described in Section 6.3.2.

Now follows a more detailed description of the functionality and architecture of all three parts.

For each clock cycle, the sum of three adjacent channels is built for each channel. An incoming data value is accepted as an interesting data point if two conditions are met:

(i) The amplitude of the left, A_l, and right, A_r, channels are smaller than the middle channel A_c ($A_l < A_c$ and $A_r \leq A_c$).

(ii) The sum of these three amplitudes is greater than a configurable threshold th, i.e. $A_l + A_c + A_r > th$, which can be set to full scale in order to mask any given channel.

This first block operates at the sampling frequency of the ADCs of about 10 MHz. Note that in order to keep all internal clock frequencies as low as possible (ADC clock), more logic than absolutely necessary is used here. The arithmetics is fast enough, easily supporting a $4\times$ or higher multiplexing, which is not implemented in order to avoid higher clock frequencies and related digital noise.

If a three-channel amplitude group meets the specified conditions, a four-parameter block (A_x, A_s, ID, hit) is queued for further calculations. It contains the amplitude A_x of the larger of the left or right neighbors of the current channel, the sum amplitude A_s of these three channels, the channel number ID, and a flag hit that indicates that the data from this channel form a valid candidate to compute further fit parameters. If no channel complies with the hit conditions, nothing is calculated or stored in the register file (FIT). For each valid time bin, the acquisition kernel computes the following set of parameters : x_i, y_i, $x_i \cdot y_i$, x_i^2, y_i^2, hc, trd. The parameter x_i represents the position in drift direction and is encoded as a sampling time bin number. The parameter y_i is the location of the charge cloud in the given time bin in y direction, which is determined using two lookup tables, encoding the pad response function and providing two independent measures of the position. The comparison of the two ratios with the tabulated expectation based on the pad response function allows at this stage also to reject merged hits. The parameter hc is a flag that for a valid group is one and the update of which in FIT encodes in the end the number N of space points for a *tracklet* (see equation 6.2). The parameter trd reflects the amplitude of the group.

In the third part, the calculated parameters are used to update a read-modify-write memory (FIT). In each cycle, a full parameter line is read out and the sums of the calculated parameters are written back to the memory. For each data channel, a memory line for all parameters is implemented. The fit register is also the interface to the MIMD processor, which works on these calculated data after the drift-time. While the TPP works in acquisition mode, the MIMD processor is in sleep mode with its clocks disabled. At the end of the drift-time, if parameters were calculated, the TPP wakes up the MIMD.

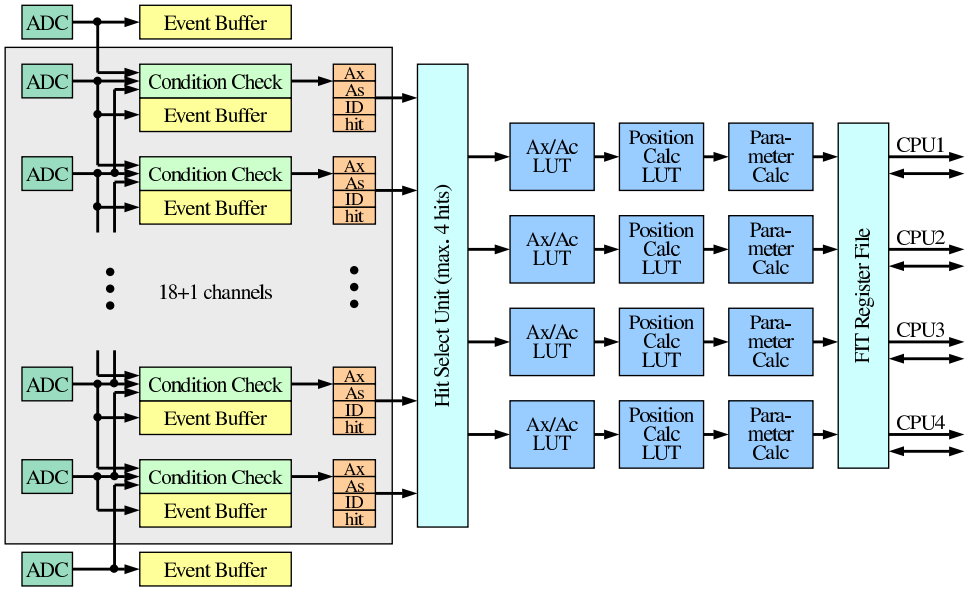

Figure 6.5: Schematic block diagram of the TPP chip.

The TPP block diagram is shown in Fig. 6.5. Altogether, the TPP comprises the following building blocks : 19 data channels with event buffers capable to store the ADC values; a cluster finder to select

interesting data and sum three channels ('Condition Check'); a unit that selects up to four clusters (largest amplitudes) for further processing ('Hit Select Unit'); a position calculation block to calculate the fit input parameters; and a read-modify-write memory block to store the calculated data in the register file ('FIT') which is also the interface to the MIMD processor. Note that the 19 channel TPP includes 2+1 additional channels of its neighboring MCMs and thus from different preamplifier chips (refer to section 6.3.1), which are already digitized and stored at these neighboring MCMs. In order to verify the gain matching between the different chips these shared channels are archived also within the TPP. In addition to the basic functional blocks, each channel has a configurable block to subtract a pedestal and apply a threshold. Each block contains two fast multipliers, as provided by the library of the silicon synthesizer, and one special divider that is implemented as a lookup table. The special divider actually approximates the division by transforming the data to a logarithmic scale. This is realized by generating logic from lookup tables and adjusting the precision of the calculation to the quality of the measurement process. The next chapter describes the building blocks in more detail.

Front End

The front-end works with a frequency of about 10 MHz, and provides the interface to the integrated ADCs. Each chip receives data from 21 ADCs, and each data channel receives data from three ADCs: from the current ADC, and from its left and right neighbor. First, a multiplexer selects the larger value out of the right and left channel; this choice is made by a comparator block (see Fig. 6.4). Then the ratio to the center channel is built and the position y_i is calculated. Both, the normalization and the position reconstruction can be performed by programmable look-up tables.

Configuration

The TPP allows for individual configuration of pedestal, thresholds and drift length. The pedestal value can be configured for each channel in the TPP. The pedestal is 10 Bit wide, thus allowing for the elimination of defect channels. The second is the 12 Bit wide threshold that is used to check the th condition in the TPP. It also serves for the zero suppression that eliminates noise. The drift length, a 6 Bit number, is necessary to configure the address counter of the event buffer. The event buffer starts to fetch data when the pretrigger is detected in the chip and stops when the address counter reaches the drift length. This is common for all 19 channels in a chip.

The TPP configuration is done by the MIMD processor in a programmed I/O fashion, when in the configuration state. A device is added to the I/O memory space supporting the TPP configuration functionality. In order to keep on-chip routing simple and to avoid complex configuration buses the TPP configuration is shifted in or out sequentially, allowing to implement the configuration registers as shift registers. There is, however, a shadow register allowing to (re)define the actual configuration in one clock by copying to/from the shadow register. In order to set the configuration shift registers efficiently, one of the MIMD CPUs writes to the TPP configuration logic. As a consequence an internal state machine shifts the received 32 Bit wide configuration word sequentially into a shadow shift register of the real configuration register. The MIMD processor continues until all bits have been shifted in. The number of bits in the shadow configuration register are always multiple of 32 bits. The shadow register is then copied into the actual configuration registers by asserting an appropriate strobe signal, which is triggered by writing to another defined region within the TPP configuration block. Configuration reads are implemented equivalently, with the exception that the shadow configuration register is copied first from the actual configuration registers and then read out by the MIMD processor, reading 32 bits per transaction.

The configuration parameters are taken from the MIMD processors global data RAM. They can be uploaded during the LTU configuration using the serial configuration link. This link is terminated in a configuration unit, having access to both the I/O bus and the global data RAM, using the address and data bus of CPU 1 as sketched in Fig. 6.6 (see also section 6.3.3.6). However, the serial configuration

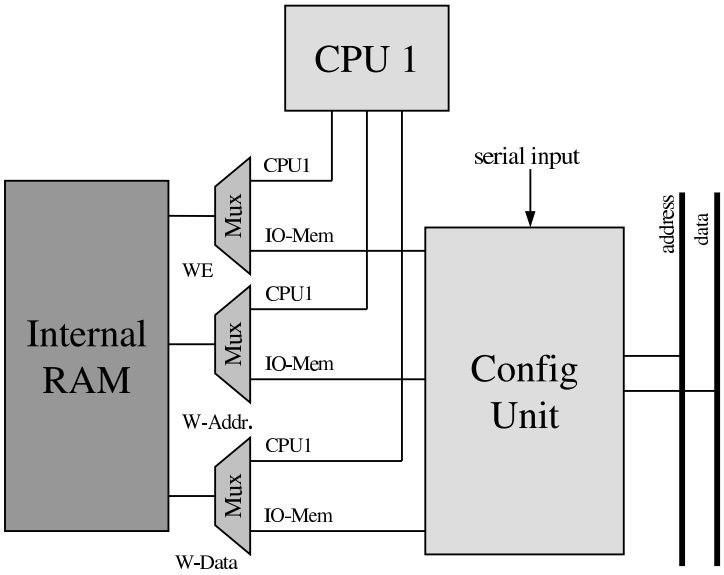

Figure 6.6: Configuration of internal RAMs.

unit can also write directly to the memory mapped TPP configuration registers mocking the programming functionality of the MIMD processor, supporting external configuration of the TPP without the assistance of the MIMD processor.

Different operating modes

The TPP has three operating modes. By default, if not idle, it is in acquisition mode to receive data from the ADCs and calculate all fit parameters. In the second mode, the event buffer is accessible by the MIMD processor. Since there is no explicit store instruction for the event buffer, the ports are accessed via the global I/O memory region and will be controlled by a dedicated register in the global I/O memory. This mode will be used for self test functionality of the LTU. In the third mode, the TPP reads its input from the event buffer rather than from the ADCs. In this mode an event can be replayed or a known simulated event can be downloaded and run through the system in order to verify system integrity. If chip real estate permits, it is planned to implement the event buffers deep enough in order to allow having always a backup or stored event on-line, which allows to inject a simulated event with zero dead time.

TPP prototype chip FaRo 1

The first prototype (FaRo 1) has been produced using the AMS[1] 0.35 μm CMOS process. This prototype has eight data channels. The front-end is designed to work externally at 20 MHz and internally at 80 MHz. The resulting chip integrates about 50k gates and has a core size of 14.4 mm ^{2}. A test board generating the required external clocks has been produced. As a pattern generator, an appropriate FPGA was used, which implements a PCI core, thereby allowing access via PCI. The test vectors are generated by software and downloaded through the PCI bus into the internal memory of the FPGA. On a trigger signal, the FaRo chip is subjected to the previously downloaded test data and subsequently performs the processing that takes place during the data acquisition time. Following this, the FPGA generates read out signals and collects the results from the prototype. Then all data are shipped through the PCI bus into a Linux PC's memory and are compared with the expected data. The FaRo 1 chip was validated by testing it with more than three million test vectors. The set of test vectors was designed to cover critical areas of

[1] Austria Micro Systems, *www.amsint.com*

the design. In addition, randomly generated test vectors have been used. During the tests, no errors were detected. The FaRo 1 test-board with the Orca FPGA board is shown in Fig. 6.7.

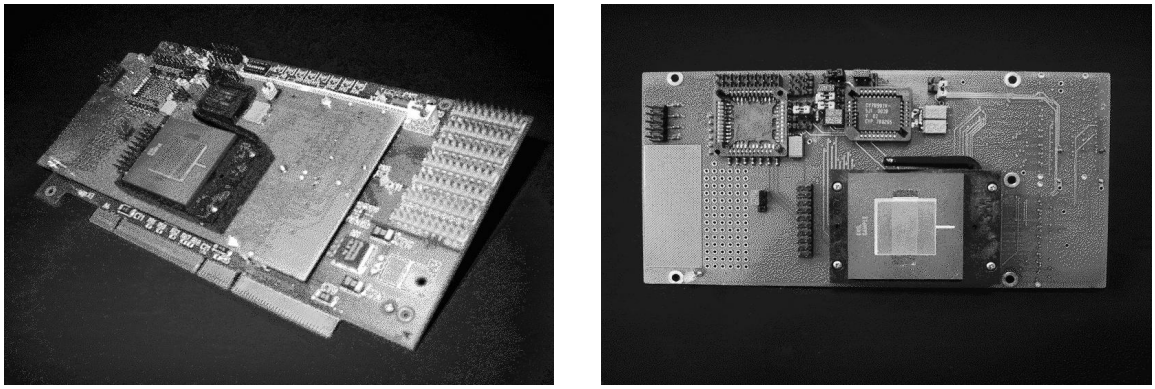

Figure 6.7: FaRo test board.

Calculation of the power consumption of the FaRo 1 TPP is done with a separate power supply for FaRo in the test setup. A fatigue test shows that the power consumption is only dependent on the clock frequency. One channel in the whole architecture has a power consumption of about 100 mW. The test chip works internally with a fourfold clock frequency compared to the front-end. The test results are shown in Fig. 6.8.

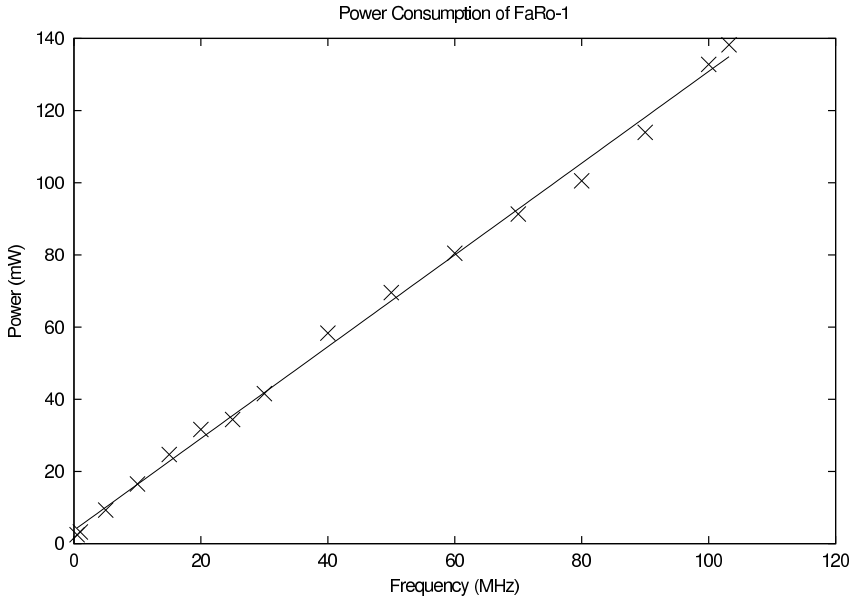

Figure 6.8: Power consumption of the FaRo 1 chip.

6.3.3 Tracklet processor

This chapter focuses on the architecture and functionality of the MIMD trigger processor which is integrated into the read out system and will be implemented on the MCM that is described in Section 5.8.

6.3.3.1 Architecture and functionality

The architecture allows the concurrent execution of multiple threads on a shared memory and provides an efficient means for inter-thread communication and synchronization. Four Harvard-style CPUs are

closely coupled by sharing the same data and instruction memory. In addition, a subset of their registers is globally accessible.

The task of the system is to execute flexible trigger code in less than 1 μs . As described, the TTP digitizes and captures the amplitudes of 18 channels and executes a hard-coded algorithm, which provides a first selection of up to four track candidates and calculates parameters for the linear fit.

This processing stage is performed by a four-node MIMD processor. Each node implements a private register file and a global register file to share parameters and provide a means for synchronization. To simplify the decode/fetch phase, all arithmetic instructions work register-to-register. The size of an instruction is limited to a single 24 Bit word. The Harvard-style architecture foresees only two pipeline stages : one fetch/decode stage and one execute/write back stage. The data and program stores are implemented as shared internal quad-ported RAMs. The data RAM is accessible via load and store operations. Data from the acquisition stage of the TPP are retrieved by a dedicated load instruction. The instruction set is RISC-like. In addition to the common arithmetic and logical operations, instructions have been added to handle synchronization between the CPUs. An overview of the architecture is sketched in Fig. 6.9.

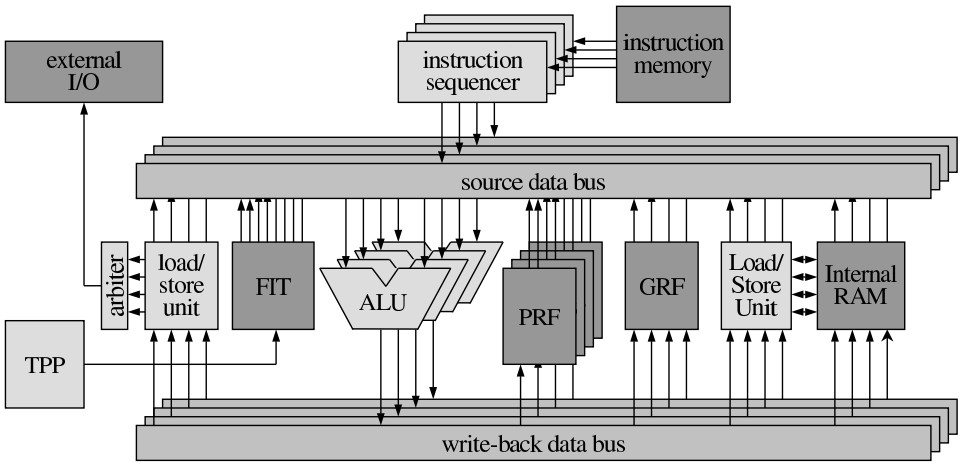

Figure 6.9: Schematic block diagram of the MIMD processor.

Most instructions can be executed in two clock cycles. The four independent CPUs work on the data from the TPP to execute four independent trigger algorithms simultaneously. Due to the read-modify-write capability of the global registers, data conflicts in the pipeline are avoided. This allows for the production of highly efficient code for the expected computations. First versions of the code indicate that, in many cases, the results from one instruction are used during the next cycle. In deeply pipelined RISC architectures, the results must be forwarded using dedicated data paths. As an alternative, no-operation instructions (NOP) can be inserted into the instruction stream. This architecture avoids both, the hardware and software the complications resulting from data dependencies.

The interface to external components is realized via the private register file. The CPU also supports synchronization between the nodes by a synchronization register file. Writing to those registers creates signals which, together with some additional instructions, provide a versatile means of synchronization.

Each node has a 32 Bit wide data path. Thus, for those arithmetic elements employing more than 32 Bit wide operands/results, multiple registers have to be used. As can be seen in Section 6.3.3.2, only three operands can be addressed by one instruction. Thus, for the upper part of the results, a dedicated register is used. Each implemented data memory is 32 Bit wide. The processor has four independent instruction sequencers. The instruction memory is a full custom 24 Bit wide quad-ported memory that is implemented as internal RAM with 2048 entries. The first test program requires about 200 entries (excluding the zero suppressing readout program) and we estimate that 2048 is enough for

all enhancements. Each node has a separate decode block to decode the current instruction. The decoded instructions are stored in a pipeline register. In every cycle, the nodes can fetch data from one of the following sources : the private register file (pRF); the global register file (GRF); the fit register file (FIT) containing the results of the TPP; the internal RAM; the global I/O memory; and the event buffer from the TPP. To allow simultaneous access, the interface to the TPP is implemented as a multi-ported register file (FIT) with a fixed assignment of a given *tracklet* candidate to a CPU, which is determined by the TPP at the end of the drift time. Results can be written back either to the PRF or the GRF. It is expected that a node works mainly on its PRF. To exchange data between nodes, the GRF or the RAM can be used. To access the RAM, load/store instructions are used. The GRF has to provide four write and four read ports to allow concurrent access by all the nodes. It has proven to be convenient to keep certain constants in read-only registers. In this architecture, we have foreseen some constant values. These constants are implemented by using two of the bits that encode the source register block. This mechanism can be used to introduce up to 16 constants.

6.3.3.2 Instruction set and format

The RISC instruction set implements fixed-length 24 Bit wide instructions. Four major addressing modes are supported : immediate, register direct, register indirect, and memory direct. Figure 6.10 sketches the seven different supported instruction formats.

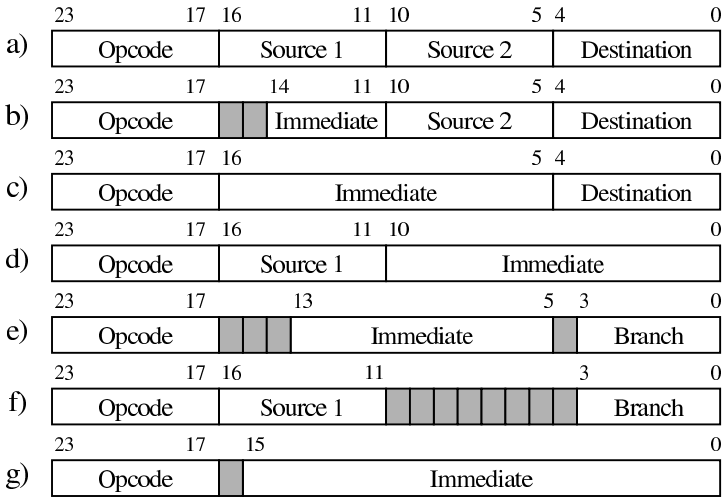

Figure 6.10: Instruction formats.

The instruction code is represented by a 7 Bit wide field to allow sufficient room for additional instructions. Currently, 70 instructions are implemented. The supported instructions sketched in Fig. 6.10 are in the given order :

- Arithmetic, logical, rotate, move, compare, interrupt instructions, instructions for synchronization, and register indirect load/store instructions
- Shift instructions
- Immediate Move and Load
- Immediate Compare and Store
- Immediate Branch instructions
- Register indirect jumps

- Special instructions for synchronization and interrupts

Most of the instructions, known from general purpose processors are supported. However, there is one extra group of instructions for synchronization between the CPUs.

The instruction set has no explicit instruction for subroutine calls. However, any register can be used as stack pointer and the program counter can be linked, thus allowing the implementation of subroutine calls. The whole instruction set is shown in Table 6.1.

6.3.3.3 Quad Ported Memory

A static RAM cell was designed to be used in the TRD trigger system. In the MIMD processor, it will serve as instruction memory and internal RAM to provide access for the four CPUs in each clock cycle. The shared instruction memory allows considerable savings in die area. Since the static RAM is as fast as registers are, the system design is simplified.

To minimize chip area and power consumption, both of which are critical for this trigger processor, a full custom macro cell has been developed in the AMS 0.35 μm CMOS process with three metal layers and a VDD voltage of 3.3 V. It is a scalable memory block with a maximum block size of 64 lines and a maximum line width of approximately 60 Bit. Special emphasis has been put on access time and chip space utilization. Both implemented memories are organized in blocks of 64 lines. The instruction memory includes eight multiplexed blocks with 24 Bit per line. The data memory consists of four blocks with 16 Bit per line. All data ports are bi-directional.

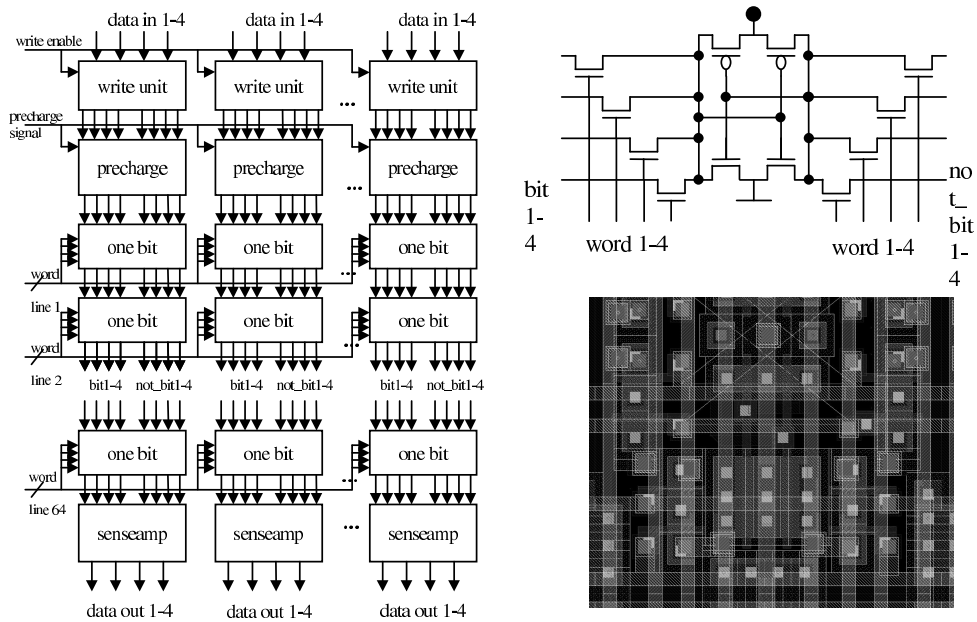

Figure 6.11: Block diagram of a memory block, as well as a schematic and layout view of a one-bit cell.

The structure of one memory block is shown at the left side in Fig. 6.11. Each block includes an independent set of address decoders (not shown), a write unit, a precharge unit, and a sense amplifier for each port, which allows them to operate asynchronously. In addition, a block contains an array of 64 lines of SRAM bit cells.

A single bit cell, shown in Fig. 6.11 (right panels), consists of two cross-coupled inverters. Each inverter is made up of one PMOS and one NMOS transistor. The PMOS transistor is connected to VDD and pulls the output potential to VDD if the input is on ground level. The NMOS transistor is connected to ground and pulls the output to ground if the input is VDD. The output of the first inverter is the input of the second and vice versa. This system has two states. If the input of inverter one is at VDD, it pulls the

Table 6.1: Instruction set of the MIMD processor.

No.	Opcode	Source 1	Source 2	Destination	Description
0	NOP	-	-	-	No Operation
1	ADD	PRF, FitReg	GRF, PRF, FitReg	GRF, PRF	C = a + b
2	ADC	PRF, FitReg	GRF, PRF, FitReg	GRF, PRF	C = a + b + carry
3	SUB	PRF, FitReg	GRF, PRF, FitReg	GRF, PRF	C = a - b
4	SBC	PRF, FitReg	GRF, PRF, FitReg	GRF, PRF	C = a - b - carry
5	MUL	PRF, FitReg	GRF, PRF, FitReg	GRF, PRF	C = a * b
6	MUS	PRF, FitReg	GRF, PRF, FitReg	GRF, PRF	C = a * b
7	DIV	PRF, FitReg	GRF, PRF, FitReg	-	C = a / b
8	DIE	-	-	-	C = a / b
9	AND	PRF, FitReg	GRF, PRF, FitReg	GRF, PRF	C = a & b
10	ATT	PRF, FitReg	GRF, PRF, FitReg	GRF, PRF	a & b
11	ORR	PRF, FitReg	GRF, PRF, FitReg	GRF, PRF	C = a | b
12	COM	-	PRF, FitReg	GRF, PRF	C = ! a
13	NEG	-	PRF, FitReg	GRF, PRF	C = (! a) +1
14	EOR	PRF, FitReg	GRF, PRF, FitReg	GRF, PRF	C = a ^ b
15	SHA	implicit	GRF, PRF, FitReg	GRF, PRF	C = a shar X
16	SHT	implicit	GRF, PRF, FitReg	GRF, PRF	C = a << X
17	ROR	-	GRF, PRF, FitReg	GRF, PRF	C = ROR(a, carry)
18	MOV	-	GRF, PRF, FitReg	GRF, PRF	C = a
19	MVI	-	implicit	GRF, PRF	C = implicit
20	CMP	PRF, FitReg	GRF, PRF, FitReg	-	a - b
21	CPI	PRF, FitReg	implicit	-	a - b
22	CPC	PRF	GRF, PRF, FitReg	-	a - b - c
23	BSS	-	implicit	-	branch if s
24	JSS	PRF	-	-	branch if s
25	BSC	-	implicit	-	branch if not s
26	JSC	PRF	-	-	branch if not s
27	BZS	-	implicit	-	branch if zero
28	JZS	PRF	-	-	branch if zero
29	BZC	-	implicit	-	branch if not zero
30	JZC	PRF	-	-	branch if not zero
31	BVS	-	implicit	-	branch if overflow (V)
32	JVS	PRF	-	-	branch if overflow (V)
33	BVC	-	implicit	-	branch if not overflow (V)
34	JVC	PRF	-	-	branch if not overflow (V)
35	BNS	-	implicit	-	branch if n
36	JNS	PRF	-	-	branch if n
37	BNC	-	implicit	-	branch if not n
38	JNC	PRF	-	-	branch if not n
39	BCS	-	implicit	-	branch if c
40	JCS	PRF	-	-	branch if c
41	BCC	-	implicit	-	branch if not c
42	JCC	PRF	-	-	branch if not c
43	BBS	-	implicit	-	branch if b
44	JBS	PRF	-	-	branch if b
45	BBC	-	implicit	-	branch if not b
46	JBC	PRF	-	-	branch if not b
47	BRA	-	implicit	-	unconditional branch
48	JMP	PRF	-	-	unconditional branch
49	SYN	-	-	-	wait on ALU
50	SYT	-	-	PRF, GRF	test wait
51	SEM	-	implicit	-	Set Syn mask
52	LRA	-	PRF	PRF, GRF	Load from RAM
53	LRI	-	implicit	PRF, GRF	Load from RAM
54	SRA	PRF	PRF	-	Store to RAM
55	SRI	PRF	-	implicit	Store to RAM
56	LBU	-	PRF	PRF, GRF	Load Evt.-Buff
57	LBI	-	implicit	PRF, GRF	Load Evt.-Buff
58	LPA	-	PRF	PRF, GRF	Load from private I/O Mem
59	LPI	-	implicit	PRF, GRF	Load from private I/O Mem
60	SPA	PRF	PRF	-	Store to private I/O Mem
61	SPI	PRF	-	implicit	Store to private I/O Mem
62	LGA	-	PRF	PRF, GRF	Load from global I/O Mem
63	LGI	-	implicit	PRF, GRF	Load from global I/O Mem
64	SGA	PRF	PRF	-	Store to global I/O Mem
65	SGI	PRF	-	implicit	Store to global I/O Mem
66	CLI	-	-	-	Clear Interrupt
67	STI	-	-	-	Set Interrupt
68	INT	-	implicit	-	Software interrupt
69	IRT	-	-	-	Back form interrupt

input of inverter two down, which subsequently pulls the input of inverter one up, and by this stabilizes the system. This state can represent the digital value one. By reversing the inverters' input potential, the second state is achieved, representing the digital zero.

A single port of the bit cell consists of two minimum size NMOS transistors connecting the input of inverter one to the bit line and the input of the second inverter to the not_bit line. The gates of these pass

transistors are connected to the word line, which is driven by the address decoder as mentioned later.

This setup is used for each of the four ports. The resulting load has to be taken into account in the design of the inverters. In the worst case scenario, a single bit cell is addressed by all four ports. Then the capacity of four bit/not_bit lines has to be driven. This mainly results from the parasitic capacity of the pass transistors of all bit cells in a column. This requires a layout of the NMOS transistors of the two inverters with the threefold area of a minimum-sized transistor. The PMOS transistors can be kept at minimum size. As a result, the cell can discharge a bit/not_bit line quickly, but charging is significantly slower, of the order of a factor 10. In total, a single bit cell includes two PMOS transistors, two threefold NMOS and eight NMOS pass transistors. The cells are arranged in a rectangular grid. The power, ground and bit/not_bit lines run vertically through the cells, and the traces for the word lines are arranged horizontally.

The area of a single bit cell is determined mainly by the eight pass transistors and the routing of the four word lines and the eight bit/not_bit lines. The resulting cell is 12.9 μm wide and 9.6 μm high. The needed area is approximately 1.6 times that of a single ported SRAM cell with two pass transistors and minimum size inverters.

The number of memory lines per block is limited by the maximum capacity a cell can drive in the permitted time frame. As a compromise between large blocks and small inverters, we chose a block length of 64. A bigger inverter can drive more memory lines. To realize a 128 word block, at least fourfold-sized NMOS transistors are required. These transistors increase the effective size of a cell in such a way that the block would require more space than two 64-line blocks. The reason for this is mainly due to the fact that the threefold sized NMOS transistors can be placed in an otherwise free rectangle formed by four pass transistors. These estimates take into account the peripheral logic.

To read out a memory line through one of the four ports, all bit/not_bit lines of this port are first precharged. Simultaneously, the address decoder decodes the address. Then the address decoder drives the addressed word line. This triggers all bit cells from this word line to drive the precharged bit/not_bit lines. The sense amplifiers detect the voltage differences between these lines and write out the stored bits. To write data, the address decoder has to decode the address and drive the addressed word line. Instead of precharging, the write unit must drive bit and not_bit lines to opposite levels. Depending on the desired data value, the bit line is driven at VDD and the not_bit line is driven at ground level or vice versa.

Figure 6.12 shows the results from the simulation. In the simulation, a logical zero and a logical one is read alternating from two bit cells. The first graph (top) shows the global clock signal that triggers the pass transistors. The bit/not_bit lines are precharged between two reading cycles, shown in the second graph. To emulate the effect of the rest of the memory block, they are connected to the capacity representing the block. After all pass transistors have been enabled to simulate reading on all four ports, the bit cell needs 4 ns to discharge either the bit or the not_bit line. However, the sense amplifier needs only 1.7 ns to drive its output to 90% of VDD in case of a digital one, shown in the third graph. A standard buffer (BU2) connected to the sense amplifier's output provides the digital value 0.5 ns after the pass transistors have been enabled, shown in the last graph. The address decoder needs 1.6 ns from applying the address to enabling the word line. This time is used for precharging the bit/not_bit lines. A complete read cycle needs 2.1 ns from applying the address to the output of the digital data.

The whole memory block has also been simulated and the times mentioned above have been verified. The chips have been received from the manufacturer but testing has not completed in time for this report.

6.3.3.4 Arithmetic Logical Unit

The ALU applies arithmetic and logic operations on two integer operands. It implements binary logic (and, or, xor) and the full set of basic arithmetic operations, i.e., addition, subtraction, multiplication and division as it's used for the equations 6.1 to 6.5. In addition, bit shifts of variable distance can be applied

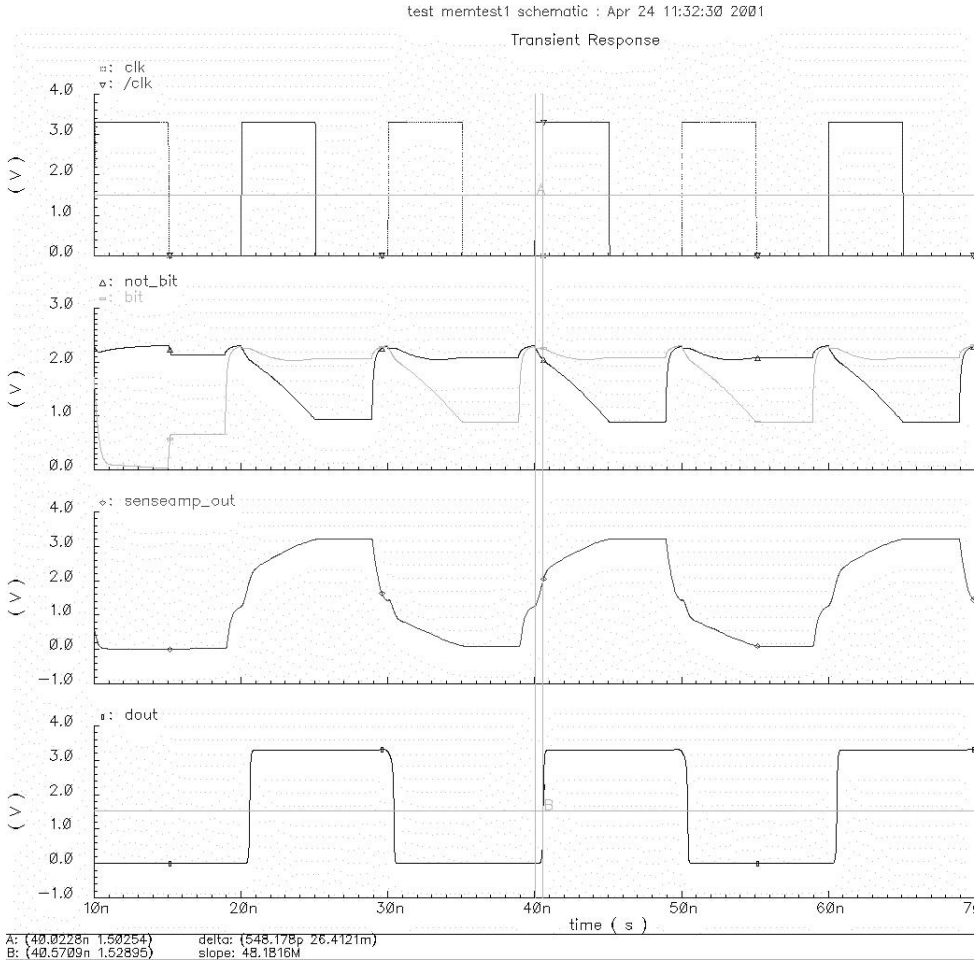

Figure 6.12: Simulation results of reading from one memory cell of the Quad Ported Memory. The address is produced with the clock signal clk. The second row shows the effected bit line and inverted bit line. The signals senseamp_out and dout show the output of the memory cell before and after the output buffer. The vertical lines at 40 ns show an access time of the core cell of less than 1 ns (plus address decode).

to one of the operands.

To allow efficient control of the program flow, the ALU generates five status flags : carry, zero, two's-complement overflow, negative, and signed. An auxiliary input port is available to allow integer division with double-width dividends. Similarly, an auxiliary output port allows double-width multiplication results. All operations can work on negative integers in two's complement representation. Operation is controlled by a 4 Bit opcode, which is specifically optimized to minimize the need for internal control logic. To improve design flexibility and reusability, the input and output data width is fully parameterizable.

The primary implementation objective was to allow for high clock rates, while still performing most arithmetic operations in a single clock cycle using a non-pipelined architecture. Further requirements were low power consumption and the possibility to optimize the implementation for different clock rates. These implementation goals suggest a modular design, thus decoupling parts for different complexity and speed and allowing to switch the implementation of a component according to specific requirements.

The ALU main module implements only the basic operations like addition/subtraction and Boolean logic. These operations are non-pipelined. Multiplication and division, being more complex operations, are performed by separate modules for which several different implementations are available. These implementations include both pipelined and non-pipelined designs. The available implementations for

the divider include a CSA-based radix-4 divider [6], which has been constructed for high clock rates. For the present version of the ALU, a data word width of 32 Bit is required. To minimize data dependency problems with most of the operations, a fully combinatorial multiplier has been selected.

6.3.3.5 Synchronization

The MIMD implements four CPUs operating independently of each other. However, a means for synchronization, wake-up from stand-by and the like, is required. Figure 6.13 shows a schematic view of the GRF and the associated flag bits in each of the CPUs, forming a private 16 Bit synchronization register. While each CPU has read/write access to all 16 GRF registers, the GRF is grouped into four sets of four registers, where each given set is assigned to a particular CPU for use as a mailbox-type register in order to implement the desired synchronization primitives. When writing to one of the GRF registers within a set that is associated to a CPU, a corresponding bit is cleared in the synchronization register of all the other CPUs. This global setting of flags, triggered by register access, is the foundation on which the synchronization is based.

Three instructions are used for synchronization : SYN, SEM and SYT. SEM sets the local synchronization register to the mask provided by the argument mask16. Then the program counter of this CPU is suspended by the SYN instruction until the mask is completely cleared by write access of the corresponding CPUs to their associated registers in the GRF. The SYT instruction copies the content of the local synchronization register to the private register specified by the pRF argument. This mechanism allows for the implementation of flexible synchronization patterns in software.

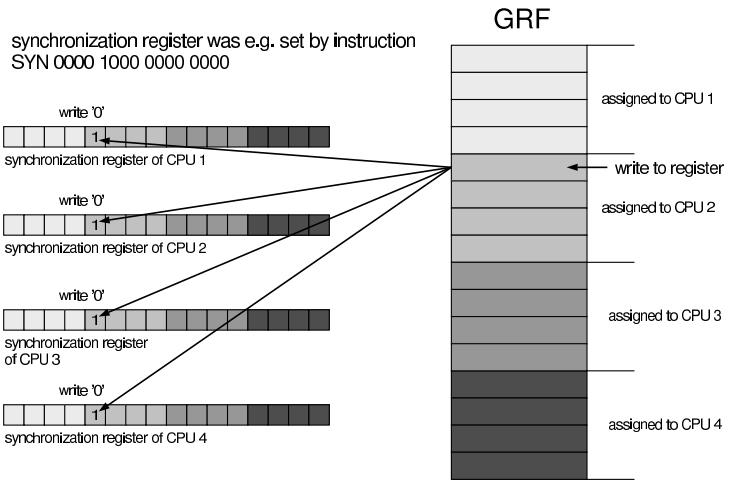

Figure 6.13: Synchronization mechanism.

6.3.3.6 Configuration

The trigger processor is configurable in the instruction memory, internal memory and some constants that are used in the trigger program. The memory that contains the used constants is 16 words deep. The currently used constants are listed in Table 6.2.

The memory of the interrupt handler from each CPU is accessible via the global I/O space. Each interrupt handler has 16 entries in the I/O space. In total 64 words are required from the memory space for the interrupt handler and can be configured by any bus master.

The internal RAM and the instruction memory are not located in the global I/O space, and thus a priori not directly accessible. Hence, it is necessary to implement a mechanism that allows to configure the internal memories. The instruction memory is programmable by a dedicated port from CPU 1. First,

Table 6.2: Entries in the constant memory.

No.	Constant
0	0
1	1
2	2
3	3
4	4
5	CPU ID (0-3)
6	chip ID
7	max. time bins
8	effective distance to the projection plane
9	unused
10	width of pad (in bins)
11	square of No. 10
12	0x8000
13	unused
14	-2
15	-1

the configuration unit (bus master) writes a specific word in the global I/O address space. Then the data and address port are decoupled from CPU 1 and switched to a hard-wired path to the global I/O address space. The memory is now accessible by reading/writing appropriate regions in the global I/O address space. This is done by presetting a start address register first and then reading or writing subsequent data words in an auto increment fashion. Finally, the data path is switched back to CPU 1, which is done by another write cycle with a dedicated word in the I/O address space (refer to Fig. 6.6). The internal RAM is configured in a similar manner.

The size of the I/O memory space is limited by the addressing scheme of the processor. Hence, the memory has a limit of 2 k words. By using an indirect addressing mode, there is no space limit. Each entry is 32 Bit wide. All connected clients at the global I/O bus have a synchronous behavior. They can only work on private data and receive data from the CPUs. It is not allowed for the clients to work on internal data of the processor.

6.3.3.7 Interface to Tracklet Preprocessor

The interface between the tracklet preprocessor (TPP) and the MIMD processor is the FIT register file. It is an eight-ported register file that is write-only by the TPP and is read-only for the MIMD processor. Each CPU has two read ports, however, each CPU has access to every line in the register. During the drift time, the TPP fills the register with the fit parameters in read-modify-write cycles. The FIT register has 19 lines with six words, and the line number corresponds to the channel number of the chip. After the drift time, up to four *tracklets* are selected. Also, the parameters from the next channels are addressable. This allows for merging of parameters from two channels without the need to increment the address register. The selection criterion is the number of accumulated hits during the drift time. A channel is selected as a stiff track candidate if it has a minimum of eight hits during the drift time. The data from two channels are merged if both channels have a minimum of four hits each during the drift time. There are three additional data channels on each LTU (refer to Fig. 5.2). In order to prevent shadow tracks, the last channel (labeled 0+) is only used to merge two channels on the given LTU between channel 17 and 0+. If the hit count in channel 0+ is larger than seven, the hits are omitted in this chip and the tracklet is calculated in the next chip.

6.3.3.8 Input/Output interface

The interface to the read out module (see Chapter 7) is located in the private I/O address space of each CPU. Each CPU can write directly to this region without arbitration logic, using a dedicated store in-

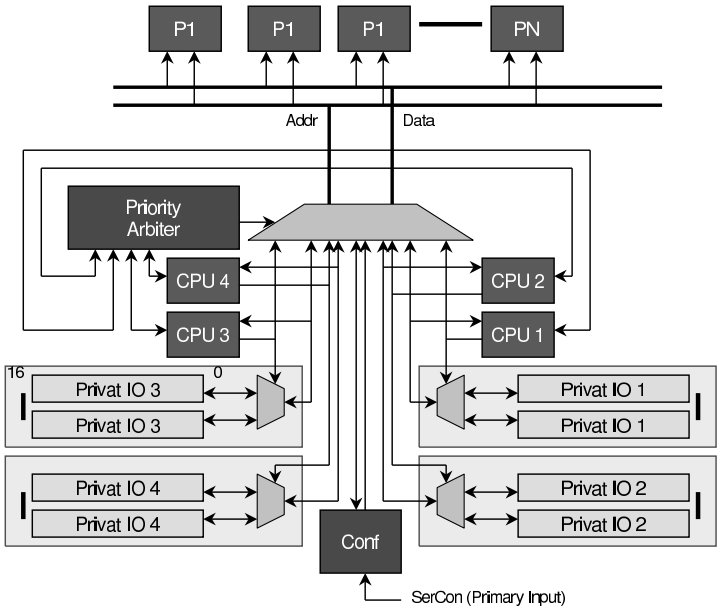

Figure 6.14: Interface to I/O address space.

struction. The interface to all other integrated peripheries is located in the global I/O address space. Only one CPU can access this I/O bus at the same time. The access is managed by a priority arbiter. The serial configuration node is a bus master and has access to the memory bus like a CPU. This enables writing data into the global I/O address space by the configuration node. It has a serial primary input port to receive data from external devices. An overview is shown in Fig. 6.14.

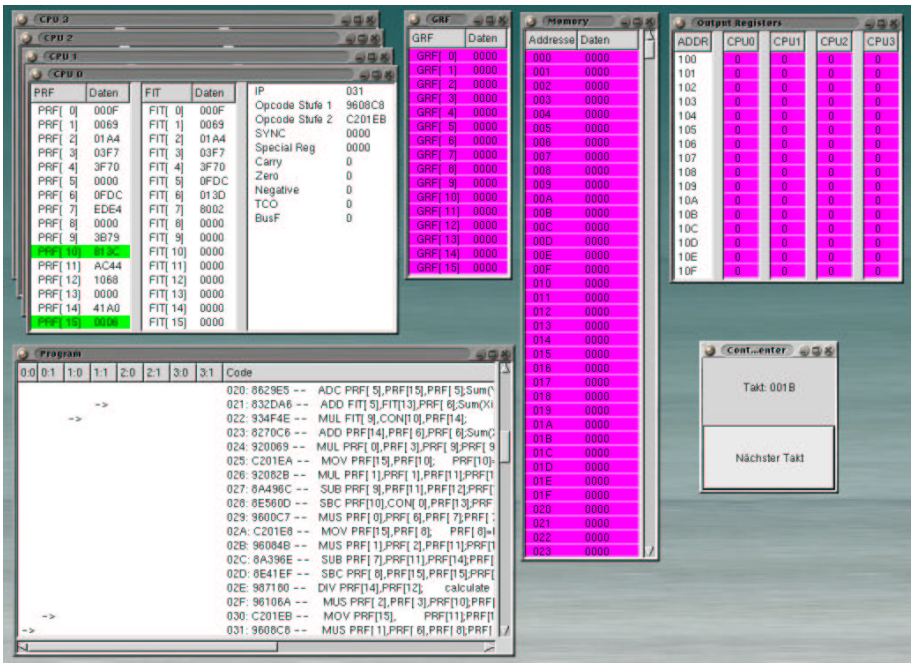

Figure 6.15: Screen-shot of the MIMD processors simulator running the trigger routine.

6.3.3.9 Interrupt

The trigger processor has eight interrupts with two priority levels. The instruction set includes four instructions to handle these interrupts : CLI, STI, INT, and IRT. CLI disables interrupts with low priority, STI allows low-level interrupts, INT is the interrupt instruction, and IRT is the return instruction from a interrupt routine. The interrupt vector table is accessible in the global I/O memory. Each CPU has sixteen entries, eight words for jump addresses and one word that provides the current level of the eight possible interrupts. The rest is yet unused. Two flag flip-flops (FF) are implemented. The first FF suspends low-level interrupts if a CLI instruction is executed or a low-level interrupt is running. The second FF suspends all interrupts if a high-level interrupt is running. Each interrupt suspends the execution of an interrupt with the same or lower priority. Basically, each interrupt checks the current priority level and the suspend flags, then stores the next program counter and jumps to the selected interrupt address. A low-level interrupt will not be executed until the decode stage contains no CLI or branch instruction. After execution of the interrupt code (low priority), the processor restores the program counter and continues the program.

6.3.3.10 Trigger and read out program

The given latency requirement allows for about 150 instructions to perform the trigger algorithm, which is an assembler routine. The data read out is not constrained much by latency, and it does not require very complex software either. Therefore, no high-level language support, such as C or C++, is expected. In order to facilitate the software development, particularly with respect to the multi-threading capability of the processor, an emulator was developed, which allows simulation of the whole processor including all its states and internal registers. Figure 6.15 shows a screen shot of the processor while executing instructions of the trigger program with raw data taken from the slow simulator described in Section 6.4.

The emulator is also being used to validate the maximum trigger processing latency. The trigger program shown in Tables 6.3 and 6.4 calculates the slope, intercept and the variance of a stiff track. Also, the intercept is projected onto the middle plain of the detector. The calculated data are stored in the private I/O memory of each CPU, which provides the interface to the track merger module, feeding the readout tree to the global tracking unit (GTU). In the worst case scenario, the algorithm will take 101 clock cycles, which correspond with a latency of 0.84 μs with a cycle time of 8.3 ns.

Unlike the trigger program, the read out program has far less stringent latency requirements, which are basically defined by the maximum available time to drain the zero suppressed raw data through the read out tree. There are many read out scenarios conceivable, which can be selected on a per-event basis. They can be dynamically changed since this is a real read out program. The baseline read out foresees zero suppression.

The power routing on the pad plane is going to be in parallel to a pad row as this is the shortest distance across the pad plane. Therefore, the eight MCMs per pad row are supported by one power strip. It has to be taken into account that the power of the MIMD processors cannot be supported by the power supplies for all CPUs simultaneously. In the case of a trigger program, those processors are powered by appropriate buffer capacitors next to the digital chip. Therefore, the read out sequence is scheduled such that, at any given point in time, only one CPU per pad row is active so as not to overload the power supply rails. However, given 12 - 16 pad rows, there is still enough parallelism in the read out to guarantee saturation of the read out link.

The read out is performed in programmed I/O fashion, reading all raw data from the event buffers and storing a pre-formatted zero suppressed event fragment in the internal global data RAM, from where the data are then fetched for shipment to the read out tree.

Table 6.3: Trigger assembler routine of the MIMD processor.

#	Label	Instruction	Operands	Description
				Test FIT[7] where to start
1	Start:	ATT	FIT[7],CON[12]	MSB is set if Faro is ready with his data
2	₋Start:	BZS	₋Start	
3		ATT	FIT[7],CON[12]	MSB is set if Faro is ready with his data
4		ATT	FIT[7],CON[2]	Bit 1 is Set if CPU has to calculate data
5		BZS	End	
				Copy or merge Data from FIT-Registers to PRF – given are: FIT[0...7]: N, $\sum x_i$, $\sum y_i$, $\sum x_i^2$, $\sum y_i^2$, $\sum x_i y_i$, b, Flags FIT[8] - FIT[15] the same for the next pad TO: PRF[0...6]: N, $\sum x_i$, $\sum y_i$, $\sum x_i^2$, $\sum y_i^2$ LSW, $\sum y_i^2$ MSW, $\sum x_i y_i$
6		ATT	FIT[7],CON[1]	Test whether to merge or not
7		BZC	merge	Merge if Bit 0 is set in FIT[7]
				Copy data
8	nmerge:	MOV	FIT[0],PRF[0]	
9		MOV	FIT[1],PRF[1]	
10		MOV	FIT[2],PRF[2]	
11		MOV	FIT[3],PRF[3]	
12		MOV	FIT[4],PRF[4]	
13		MOV	CON[0],PRF[5]	
14		BRA	Linefit	
15		MOV	FIT[5],PRF[6]	
				Merge the data
16	merge:	ADD	FIT[0],FIT[8],PRF[0]	$N = N_{(1)} + N_{(2)}$
17		ADD	FIT[1],FIT[9],PRF[1]	$\sum x_i = \sum x_{i(1)} + \sum x_{i(2)}$
18		ADD	FIT[2],FIT[10],PRF[2]	
19		MUL	FIT[8],CON[10],PRF[3]	$N_{(2)} * 127$
20		ADD	PRF[2],PRF[3],PRF[2]	$\sum y_i = \sum y_{i(1)} + \sum y_{i(2)} + N_{(2)} * 127$
21		ADD	FIT[3],FIT[11],PRF[3]	$\sum x_i^2 = \sum x_{i(1)}^2 + \sum x_{i(2)}^2$
22		MUL	FIT[8],CON[11],PRF[14]	
23		ADD	PRF[14],FIT[4],PRF[4]	
24		ADC	PRF[15],CON[0],PRF[5]	$\sum y_i^2 = \sum y_{i(1)}^2 + N_{(2)} * 127^2$
25		ADD	PRF[4],FIT[12],PRF[4]	$\sum y_i^2 = \sum y_{i(1)}^2 + \sum y_{i(2)}^2 + N_{(2)} * 127^2$
26		ADC	PRF[5],CON[0],PRF[5]	
27		MUS	FIT[10],CON[10],PRF[14]	
28		SHT	1,PRF[15],PRF[15]	
29		ATT	PRF[14],CON[12]	
30		BZS	not1	
31		SHT	1,PRF[14],PRF[14]	
32		ORR	PRF[15],CON[1],PRF[15]	
33	not1:	ADD	PRF[4],PRF[14],PRF[4]	
34		ADC	PRF[5],PRF[15],PRF[5]	$\sum y_i^2 = \sum y_{i(1)}^2 + \sum y_{i(2)}^2 + N_{(2)} * 127^2 + 2 * 127 * \sum y_{i(2)}$
35		ADD	FIT[5],FIT[13],PRF[6]	$\sum x_i y_i = \sum x_{i(1)} y_{i(1)} + \sum x_{i(2)} y_{i(2)}$
36		MUL	FIT[9],CON[10],PRF[14]	
37		ADD	PRF[14],PRF[6],PRF[6]	$\sum x_i y_i = \sum x_{i(1)} y_{i(1)} + \sum x_{i(2)} y_{i(2)} + 127 * \sum x_{i(2)}$
				Linear Fit: PRF[0...6]: N, $\sum x_i$, $\sum y_i$, $\sum x_i^2$, $\sum y_i$ LSW, $\sum y_i$ MSW, $\sum x_i y_i$
38	Linefit	MUL	PRF[0],PRF[3],PRF[9]	$PRF[9] = N * \sum x_i^2 low$
39		MOV	PRF[15],PRF[10]	$PRF[10] = N * \sum x_i^2 high$
40		MUL	PRF[1],PRF[1],PRF[11]	$PRF[11] = (\sum x_i)^2$
41		SUB	PRF[9],PRF[11],PRF[12]	$PRF[12] = N * \sum x_i^2 - (\sum x_i)^2 (denominator) low$
42		SBC	PRF[10],CON[0],PRF[13]	$PRF[13] = N * \sum x_i^2 - (\sum x_i)^2 (denominator) high$
43		MUS	PRF[0],PRF[6],PRF[7]	$PRF[7] = N * \sum x_i y_i low$
44		MOV	PRF[15],PRF[8]	$PRF[8] = N * \sum x_i y_i high$
45		MUS	PRF[1],PRF[2],PRF[11]	$PRF[11] = \sum x_i * \sum y_i$
46		SUB	PRF[7],PRF[11],PRF[14]	$PRF[14] = N * \sum x_i y_i - \sum x_i * \sum y_i (nominatorm) low$
47		SBC	PRF[8],PRF[15],PRF[15]	$PRF[15] = N * \sum x_i y_i - \sum x_i * \sum y_i (nominatorm) high$
48		DIV	PRF[14],PRF[12]	Calculate m (32 BIT : 32 BIT)
49		MUS	PRF[2],PRF[3],PRF[10]	$PRF[10] = \sum y_i * \sum x_i^2 low$
50		MOV	PRF[15], PRF[11]	$PRF[11] = \sum y_i * \sum x_i^2 high$
51		MUS	PRF[1],PRF[6],PRF[8]	$PRF[8] = \sum x_i * \sum x_i y_i low$
52		MOV	PRF[15], PRF[9]	$PRF[9] = \sum x_i * \sum x_i y_i high$
53		NOP		
54		NOP		
55		NOP		
56		DIE	PRF[7]	PRF[7]=m

Table 6.4: Trigger assembler routine of the MIMD processor (part 2).

#	Label	Instruction	Operands	Description
				Caution DIE writes PRF[15]
57		SUB	PRF[10],PRF[8],PRF[14]	$PRF[14] = \sum y_i * \sum x_i^2 - \sum x_i * \sum x_i y_i (nominatorb) low$
58		SBC	PRF[11],PRF[9],PRF[15]	$PRF[15] = \sum y_i * \sum x_i^2 - \sum x_i * \sum x_i y_i (nominatorb) high$
59		DIV	PRF[14],PRF[12]	Calculate b
60		SHT	1,PRF[7],PRF[14]	$PRF[14] = 2 * m$
61		MUS	PRF[14],PRF[6],PRF[10]	$PRF[10] = 2 * m * \sum x_i y_i low$
62		SUB	PRF[4],PRF[10],PRF[10]	$PRF[10] = \sum y_i^2 - 2 * m * \sum x_i y_i low$
63		SBC	PRF[5],PRF[15],PRF[11]	$PRF[11] = \sum y_i^2 - 2 * m * \sum x_i y_i high$
64		MUS	PRF[7],PRF[7],PRF[9]	$PRF[9] = m^2$
65		MUL	PRF[9],PRF[3],PRF[9]	$PRF[9] = m^2 * \sum x_i^2$
66		MOV	PRF[15],PRF[12]	Save highword
67		DIE	PRF[8]	$PRF[8] = b$
68		ADD	PRF[10],PRF[9],PRF[10]	$PRF[10] = \sum y_i^2 - 2 * m * \sum x_i y_i + m^2 * \sum x_i^2 low$
69		ADC	PRF[11],PRF[12],PRF[11]	$PRF[11] = \sum y_i^2 - 2 * m * \sum x_i y_i + m^2 * \sum x_i^2 high$
70		MUS	PRF[14],PRF[1],PRF[12]	$PRF[12] = 2 * m * \sum x_i$
71		MUS	PRF[12],PRF[8],PRF[12]	$PRF[12] = 2 * m * b * \sum x_i$
72		ADD	PRF[10],PRF[12],PRF[10]	$PRF[10] = \sum y_i^2 - 2 * m * \sum x_i y_i + m^2 * \sum x_i^2 + 2 * m * b * \sum x_i low$
73		ADC	PRF[11],PRF[15],PRF[11]	$PRF[11] = \sum y_i^2 - 2 * m * \sum x_i y_i + m^2 * \sum x_i^2 + 2 * m * b * \sum x_i high$
74		SHT	1,PRF[8],PRF[12]	$PRF[12] = 2 * b$
75		MUS	PRF[12],PRF[2],PRF[12]	$PRF[12] = 2 * b * \sum y_i$
76		SUB	PRF[10],PRF[12],PRF[10]	$PRF[10] = \sum y_i^2 - 2 * m * \sum x_i y_i + m^2 * \sum x_i^2 + 2 * m * b * \sum x_i - 2 * b * \sum y_i low$
77		SBC	PRF[11],PRF[15],PRF[11]	$PRF[11] = Sum(Yi) - 2 * m * \sum x_i y_i + m^2 * \sum x_i^2 + 2 * m * b * \sum x_i - 2 * b * \sum y_i high$
78		MUS	PRF[0],PRF[8],PRF[12]	$PRF[12] = N * b$
79		MUS	PRF[12],PRF[8],PRF[12]	$PRF[12] = N * b^2$
80		ADD	PRF[10],PRF[12],PRF[14]	$PRF[14] = \sum y_i^2 - 2 * m * \sum x_i y_i + m^2 * \sum x_i^2 - 2 * b * \sum y_i + 2 * m * b * \sum x_i + N * b^2 low$
81		ADC	PRF[11],PRF[15],PRF[15]	$PRF[15] = \sum y_i^2 - 2 * m * \sum x_i y_i + m^2 * \sum x_i^2 - 2 * b * \sum y_i + 2 * m * b * \sum x_i + N * b^2 high$
82		MOV	CON[0],PRF[13]	DIV uses PRF[13] !!
83		ATT	PRF[7],CON[12]	$m <= 0$?
84		DIV	PRF[14],PRF[0]	Calculate variance (scale on N), DIV only changes Zero-Flag!!
85		BZS	mbz	
86		NOP		
87		NEG	PRF[7],PRF[10]	PRF[10]=abs(m)
88		CPI	PRF[10], 18	Max. relevant m = 17 (only 11 Bit cmp, no signextend)
89		BNC	End_	jJump to end if m¿=18
90		BRA	mgn_	
91	mbz:	NOP		
92		CPI	PRF[7], 18	Maximum m (only 11 Bit, no signextend!)
93		BNC	End	Jump to end if m¿=18
94		nop		
95		nop		
96	mgn_:	DIE	PRF[14]	PRF[14]=Variance low PRF[15]=Variance high
97		ATT	PRF[8],CON[12]	$b <= 0$?
98		BZS	bbz	
99		MOV	PRF[8],PRF[9]	
100		NEG	PRF[8],PRF[9]	
101	bbz:	CPI	PRF[9],191	$abs(b) <= 191$?
102		BNC	End	Jump to end if not
103		CPI	PRF[14],256	Test variance (11 Bit test!!)
104		BNC	End	
105		ADD	PRF[8],FIT[6],PRF[8]	Calculate "CHIP global" b
106		MUL	PRF[7],CON[8],PRF[13]	
107		ADD	PRF[13],PRF[8],PRF[8]	Projection on reference-plane
				– here we have PRF[0...8]: N, $\sum x_i$, $\sum y_i$, $\sum x_i^2$, $\sum y_i^2 LSW$, $\sum y_i^2 MSW$, $\sum x_i y_i$, m, b PRF[14]: Variance (only 16 Bit!) To Be Done: Data valid; Inform rest of the system
108		SPI	PRF[7],0x101	PIO[0]=m
109		SPI	PRF[8],0x102	PIO[1]=b
110		SPI	PRF[14],0x103	PIO[2]=Variance (only low word is interesting)
111		MOV	CON[1],PRF[15]	
112		SPI	PRF[15],0x100	Inform Data valid!
113	stop:	bra	stop	
114		nop		
				TODO: Zero suppression
115	End_:	BRA	End	
116	End:	NOP		Should be NOP!!
117		MOV	CON[0],PRF[15]	
118		SPI	PRF[15],0x100	Inform Data invalid!
				TODO: SPI end-programm, no data
119		BRA	Start	And return to the beginning
120	idle:	BRA	idle	
121		NOP		

6.3.4 Readout Scheme

The structure and the layout of the read out system of the 72000 MCMs is described in Chapter 7. From the trigger point of view it has to accomplish the transfer of the locally determined *tracklet* parameter to the central Global Tracking Unit (GTU) as fast as possible (see Fig. 6.1). The list of transferred bits per *tracklet* is given in Table 7.1.

6.3.5 Implementation of the Global Tracking Unit

All *tracklets* determined by the local tracking units are shipped to a global tracking unit for final trigger decision. This is implemented as a readout tree. The trigger data of a stack of six chambers are collected in the GTU. The GTU implements a Track-Matching-Unit (TMU) per phi sector. There will be no high p_t tracks traversing between sectors. Each TMU tries to find stiff tracks as shown in Fig. 6.31. This can be done logically by appropriate histogramming the projected tracklets. The GTU will be implemented in large scale FPGAs to guarantee a flexible and massive parallel implementation.

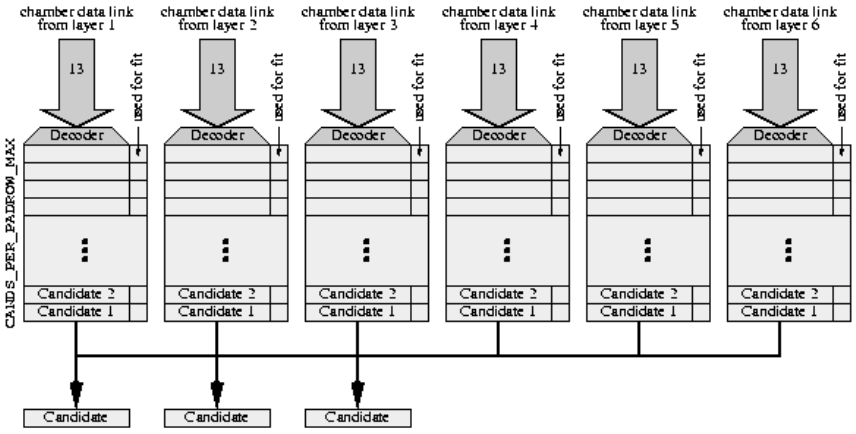

Figure 6.16: Implementation of the track matching units (TMU).

The structure shown in figure 6.16 takes into account that the data of the respective chambers arrive in a defined sequence and candidates of a track are in the same column. Therefore the processing can start as soon as the first *tracklets* arrive. All candidates of a column are stored in and accessed from a table. The number of candidates per column is fixed. The entries for chamber 1-4 are compared with all entries of all other tables. If an accumulation is found, these entries are marked, so that they cannot be used several times. It is sufficient to apply this method to only four of the six tables since a track must consist of at least three candidates. Therefore the resulting GTU architecture is a massively parallel, systolic FPGA processor performing as many as possible of such tracklet comparisons in parallel.

6.4 Simulation

The trigger concept relies heavily on the fact that a local *tracklet* search can be performed efficiently (step 1 of the trigger sequence shown in Section 6.2). The basic idea is to feed the data from consecutive time bins into pipeline ADCs and perform an online analysis of the digitized data in order to determine the inclination of the track segments with respect to the direction towards the nominal interaction vertex. All possible complications and distortions (as discussed in Chapters 5 and 11) like the E×B effect in the electron drift, the pad response function of the readout chambers, the time response function of the signal generation mechanism and of the electronics, and the differential nonlinearities of the digitization process must be taken into account quantitatively. The first step is performed in the Local Tracking Units (LTU). Due to the massively parallel processing the number of units is very large (≈ 70000) and therefore consuming most of the resources in material thickness, power and money. The number of independent channels must be optimized to obtain an acceptable signal to background ratio at affordable cost.

The modeling of the trigger response was done making use of the AliRoot environment. It allows for full event simulation employing different event generators and was used to study the occupancy, efficiency, and background performance of the envisioned trigger scheme. For a detailed description of the TRD simulation environment see Chapter 11.

Table 6.5: Parameter of the trigger simulation.

event generator	HIJING-param + signal electrons
multiplicity	varying: 400 $<$ dN/dy $<$ 8700
digitization accuracy	10 Bit
number of time samples	15 - 30
signal pulse height for minimum ionizing tracks	channel 40 (for cluster)
Signal to Noise ratio	30
Time response function	ON (as in Fig. 11.8) / OFF
Pad response function	as in Fig. 11.9
magnetic field	0.4 T

The input parameters used for the trigger simulations are listed in Table 6.5. The main objective is to find out about the most crucial parameter for detecting and selecting high momentum electrons. The effect of three quantities was investigated systematically: the event multiplicity, the digitization clock rate (number of digits) and the pulse shaping. For the latter the standard scenario described in Chapter 5 and 11 was compared to an analysis incorporating a digital filter for tail cancellation (see section 6.4.2.1) and to an academic case, where the time response function was modeled by a δ - function (TRF OFF).

To generate enough statistics for high p_t electron tracks, 200 e^+ and 200 e^- tracks were added to a parametrisation of pions and kaons called HIJING-param (see Section 12.3) with the option *genbox* of AliRoot. The momentum distribution of the electrons was chosen to be flat in the range 3-5 GeV/c .

An example of the input transverse momentum distribution for an event multiplicity corresponding to $< dN/dy >= 8500$ is shown in Fig. 6.17. Due to the added signal electrons, the effective multiplicity density for the trigger simulation is $< dN/dy >= 8700$. It should be noted again, that there is a large uncertainty about the spectral shape of the hadrons, in particular the power-law hard scattering component (see Chapter 12). The HIJING parametrisation used represents the worst case scenario. To map out the multiplicity dependence of the tracking performance the total number of primary particles emitted into the polar angle range of $35^\circ < \theta < 145^\circ$ was varied with the spectral shape in transverse direction kept as shown in Fig. 6.17.

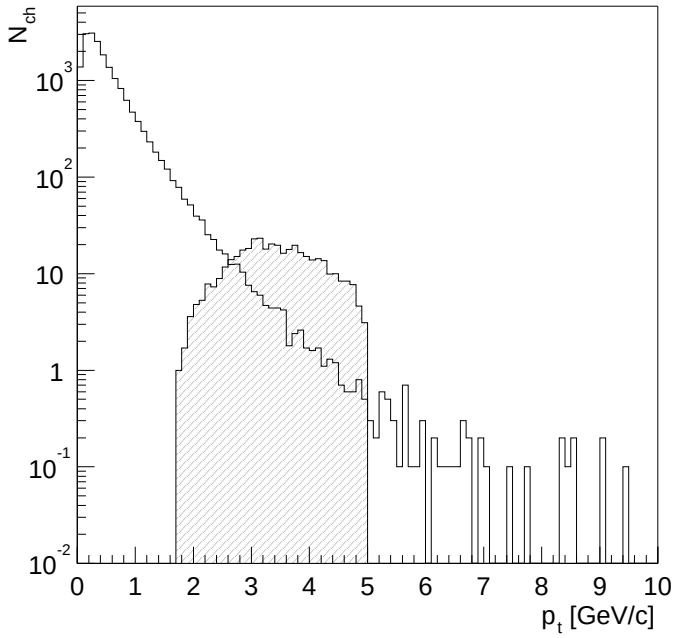

Figure 6.17: Transverse momentum distribution used as input to the trigger simulations. The hatched distribution corresponds to electrons that were added to the HIJING-param particle mix.

6.4.1 Local tracklet search

The task to be performed is visualized in Figs. 6.18 and 6.19. In Fig. 6.18 the basic quantities of the local *tracklet* search are defined. In Fig. 6.19 the digitized pulse height is shown within one pad row of one of the readout chambers of the TRD (at a polar angle of $\theta = 85^\circ$). For each time bin the pulse height is obtained after the drift of the primary electrons under the influence of the electric and magnetic field, taking into account diffusion and including transition radiation contributions. As indicated in Fig. 6.18 the position of the clusters is systematically shifted as function of the drift time due to the presence of the magnetic field. The task of the trigger system is to recognize with high efficiency stiff tracks ($p_\mathrm{t} \geq 3$ GeV/c) despite the shift due to the Lorentz angle Ψ_L. The characteristics of the interesting tracks is their small deviation from the infinite momentum limit, i.e. they have only a small angular deflection with respect to the reference line that can be constructed by connecting the point of impact with the nominal interaction vertex. As indicated in Fig. 6.18 the stiff tracks of interest occupy with the centroids of their clusters at most two neighboring pads. Charge sharing due to the pad response function distributes the signal consequently to at most 4 neighboring pads. Therefore the trigger is based on the analysis of 3 neighboring pads. For each time bin a position in pad direction (corresponding to the y - direction in the following) is determined according to the inverse pad response function that can be parameterized in a look-up table. The resulting y-positions as function of the time coordinate t_{drift} or drift distance s_{drift} are fitted by a straight line

$$y = a_0 + a_1 \cdot v_{drift} \cdot t_{drift} = a_0 + a_1 \cdot s_{drift}.$$

The fit parameters can be corrected for the effect of the Lorentz angle by the following expressions

$$a_0^{corr} = a_0 + \tan\Psi_L \cdot s_{max},$$

$$a_1^{corr} = a_1 + \tan\Psi_L.$$

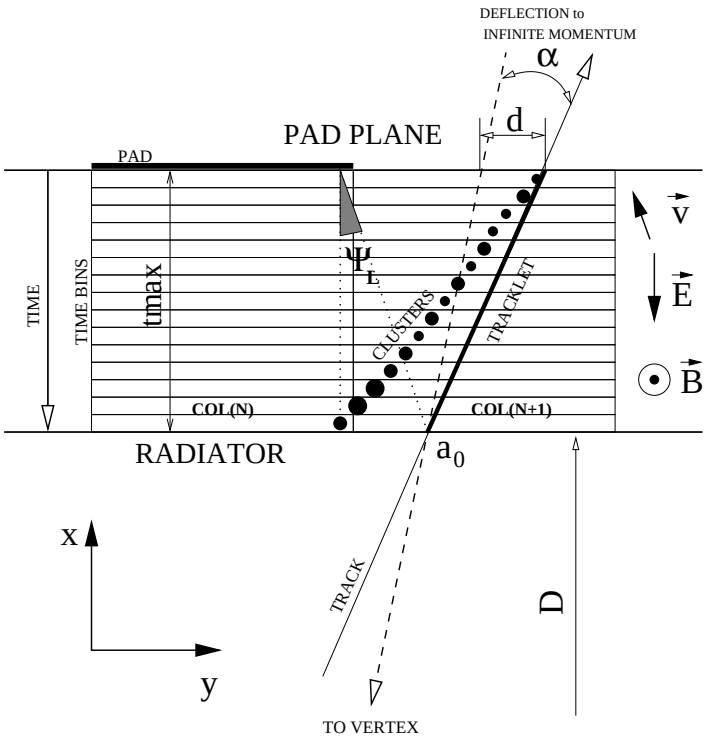

Figure 6.18: Local tracklet quantities

The linear deflection d (shown in Fig. 6.18) over the depth s_{max} of the drift region of a single chamber is given by

$$d = (a_1^{corr} - \frac{a_0}{D}) \cdot s_{max},$$

with s_{max} = 3 cm. D is the radial distance of the front surface of the readout chamber to the interaction vertex.

The angular deflection α is given by the expression

$$\alpha = \arctan \frac{a_1^{corr} - \frac{a_0^{corr}}{D}}{1 + \frac{a_1^{corr} \cdot a_0^{corr}}{D}} \approx \arctan(a_1^{corr} - \frac{a_0^{corr}}{D}).$$

Since for the tracks of interest the slope parameters are small, the second term in the denominator can be neglected (for p_t= 3 GeV/c : $|\frac{a_1^{corr} \cdot a_0^{corr}}{D}| < 0.015$).

For particles originating from the interaction vertex the deflection is related to the momentum by

$$p_t^{rec} = 0.3 \cdot B \cdot 0.01 \cdot \frac{\sqrt{(a_0^{corr})^2 + D^2}}{2 \cdot d / s_{max}} = 0.3 \cdot B \cdot 0.01 \cdot \frac{\sqrt{(a_0^{corr})^2 + D^2}}{2 \cdot \sin\alpha},$$

where all the spatial quantities are given in cm, the magnetic field B is expressed in Tesla and the reconstructed momentum is in GeV/c .

For a magnetic field of B = 0.4 T the deflection of a charged particle with a transverse momentum of 3 GeV/c over the drift range of one TRD chamber amounts to α = 3.3° or d = 1.7 mm. The linear deflection is well below the pad width of 8 mm. The trigger concept will stay valid as long as the centroids of from stiff tracks stay within a region of 3 neighboring pads, i.e. $d_{max} = 1.6$ cm or $\alpha_{max} = 28°$. The total inclination angle of the measured track has contributions from the transverse momentum ($\alpha_{max}^{p_t} = 3.3°$),

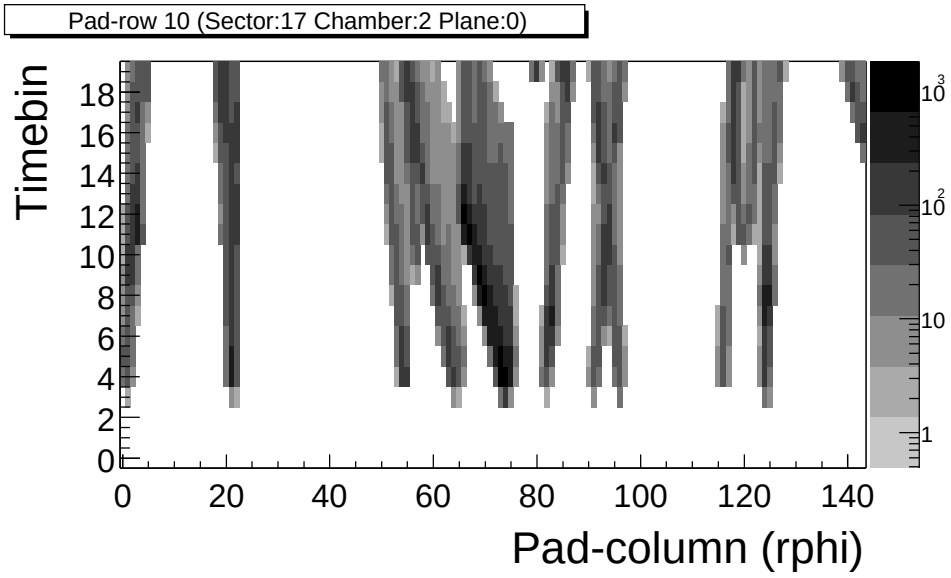

Figure 6.19: Local tracklet event display. The contour histogram shows a typical input distribution for the local tracking, i.e. the ADC contents for one pad row versus time for a full multiplicity event. 5 time bins are added before the 15 time bins sampling the drift range of the readout chamber.

the angle of incidence due to the flat surface of the chambers ($\alpha_{max}^{geo} = 10^\circ$) and from the Lorentz angle Ψ_L. The maximum allowed Lorentz angle is therefore limited to $\Psi_L = \alpha_{max} - \alpha_{max}^{p_t} - \alpha_{max}^{geo} = 14.7^\circ$. The Lorentz angle of the default gas mixture Xe,CO_2 (15%) , Ψ_L=8° , is well within the operational limits of the TRD trigger.

6.4.2 Local tracking performance

In order to achieve a sufficient resolution, the pulse height of minimum ionizing tracks has to reach a certain minimum as compared to noise and digitization errors (see Section 5.1). Due to the non-Gaussian features of the pad response function (Fig. 11.9) the most general method to calculate the position from pulse heights is realized by a look-up table (LUT). With the signal and signal to background ratio listed in Table 6.5 position resolutions of better than $\sigma_y = 400\mu$m are obtained for stiff tracks (Figs.11.12 and 11.13) and if the incidence angle is small the values are of order 200 μm (Fig. 11.14). Therefore the achievable position resolution should allow to select *tracklets* with deflections of low as d=1mm ($\alpha = 2^\circ$).

Before discussing the performance of the local tracking concept in terms of efficiency and output rate, two important configurable steps will be described in the following sections: the application of a digital filter (Section 6.4.2.1) to the preamplifier/shaper (PASA) signal and the cluster quality selection (Section 6.4.2.3).

6.4.2.1 Digital cancellation of the tail in PASA signal

The ion tail of the signal as shown in Fig. 11.8 can be parameterized by the functional form $1/(1+t/t_0)$. Such a tail can be reduced by passing the pulse through a filter (pole/zero network). The procedure is described in [7] and consists of approximating the above expression by a sum of three exponentials and adjusting the constants of the filter (resistors and capacitances in analog circuitry) such that one of them is canceled out. It has been shown [8] that a tail cancellation of equal quality can be performed on the digital signal. In order to benefit from the improvements, a scenario is considered where the functionality of the tail cancellation is included in the LTU. If it were not for the trigger performance this tail cancellation could also be performed offline.

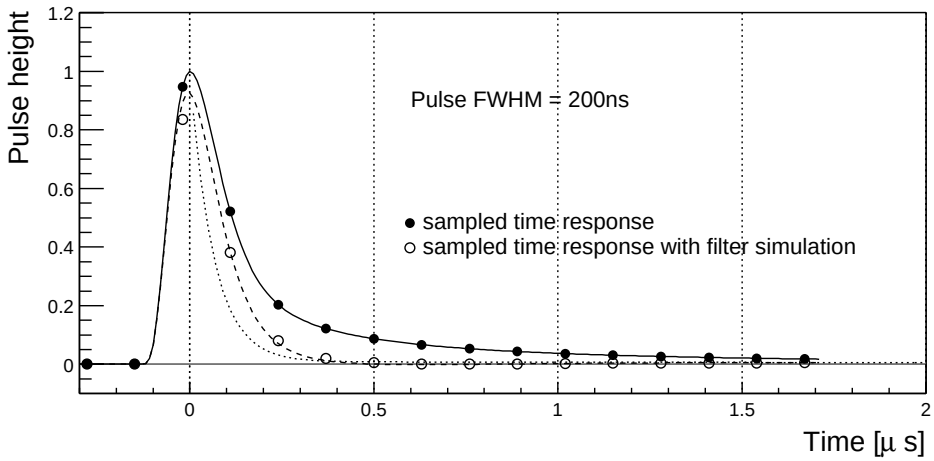

Figure 6.20: PASA signal and tail cancellation corresponding a to one pole/zero network. The solid line (filled circles) represents the input distribution (see Fig. 11.8), the dashed line (open circles) shows the response of a one pole/zero filter, the dotted curve depicts a perfect filter behavior.

In the online tracking the tail cancellation is used as an option to analyze the performance of such additional signal processing. It is implemented as a feature of the LTU algorithm in the AliRoot environment. The 15 time samples within one pad column are "filtered" using the transfer function of the pole/zero network with values calculated from [7] and adjusted to get best results. This simulates a digital implementation, superior in performance to an RC filter.

In the simulation of the Time Response Function, the PASA signal is numerically given in steps of 10 ns (the solid line in Fig. 6.20). The sampling done in 15 time bins, corresponding to 133 ns spacing, is shown in Fig. 6.20 by the closed symbols. The effect of the tail cancellation with a single pole/zero network is shown by the dashed line and the open symbols. The dotted line shows the expected tail cancellation according to the theory in [7]. With the parameters chosen and implemented in the following, the time response is almost Gaussian and extends to a maximum of 350 ns.

6.4.2.2 Occupancy

The most demanding requirement for the TRD trigger is defined by the multiplicity of the Pb-Pb collisions. The trigger concept should stay valid up to occupancies of 35% as shown in Fig. 6.21 that are expected for a rapidity density of $dN/dy = 8700$. The occupancy is evaluated with the current pad geometry described in Section 4.4 for events (generated with the event generator *HIJING-param* as described above) of different multiplicities. The occupancy for the standard scenario (i.e. 15 time bins, time response function ON, RMS width of the electronic noise equals 1 ADC channel, as described in more detail in Section 11) is given in Fig. 11.11. There as well as in Fig. 6.21 a pixel is called occupied once the pulse height exceeds the ADC channel 2. In order to save computational time the primary distributions were generated in a restricted range of polar angles ($35^\circ < \theta < 145^\circ$) leading to a reduction of the occupancy by about 14% with respect to the full calculation (solid circle as compared to the solid square at $dN/dy = 8700$). In Fig. 6.21 the average occupancy is shown as function of the event multiplicity for various additional scenarios: the influence of part of the structural material was studied by replacing the tracking medium of the space frame with air ("NO frame", open symbols). The effect of the time response function was analyzed by turning it off ("TRF off (15tb)",filled triangle up), i.e. by replacing it with a δ -function. The sampling frequency was additionally increased ("TRF off (30tb)", filled triangle down) from 15 to 30 time bins. As can be seen from Fig. 6.21 a major source of the occupancy at fixed multiplicity is the time response of the chamber/electronics that amounts to about 25% of the observed

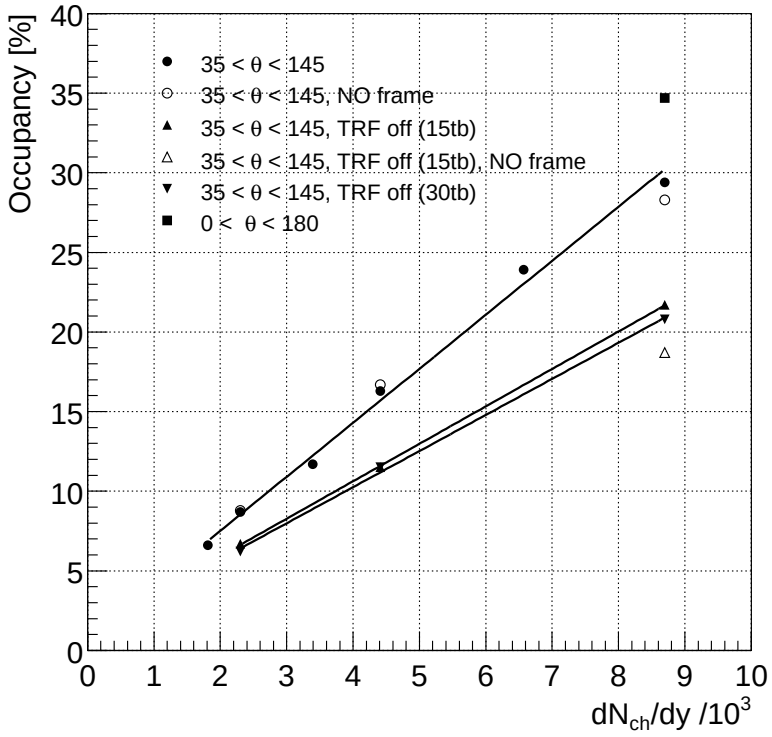

Figure 6.21: Occupancy of the TRD detector as function of multiplicity.

occupancy due to the long tails (see also Fig. 6.19). The space frame material as well as the sampling frequency have no strong influence on the occupancy.

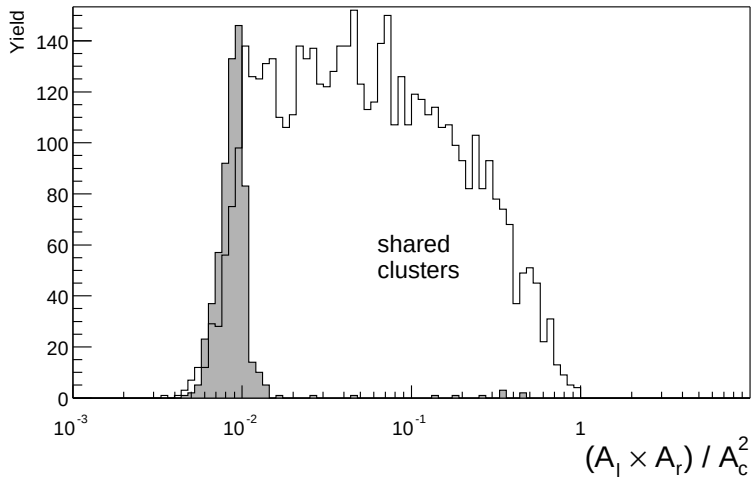

Figure 6.22: Distribution of the measure of cluster quality built from amplitude ratios ($A_l \times A_r/A_c^2$, as explained in the text) for stiff *tracklets* (p_t> 1 GeV/c) in full multiplicity events ($< dN/dy >$=8700). The grey shaded histogram is obtained for clean clusters, the solid line histogram represents clusters with contributions from more than one track (shared clusters).

6.4.2.3 Cluster quality selection

In order to achieve the best resolution (as shown below in Fig. 6.25) one has to avoid as much as possible distortion of the information by overlapping tracks (pile-up). It is therefore mandatory to inspect all contributing clusters for pile-up. This can be done by comparing the amplitude ratios of adjacent pads with the expectations for a single hit from the pad response function. The distribution of a simple measure of cluster quality, the product of the amplitudes of the side pads (A_l, A_r) normalized to the square of the amplitude of the central pad A_c, is shown in Fig. 6.22 for two cluster classes. The samples are defined with the information available within the simulation program, namely with the knowledge which tracks contributed to which cluster. The first sample (grey shaded histogram) corresponds to clean clusters, i.e. those that are generated from a single particle. The second class called shared clusters (solid line) originates from overlapping tracks. The clear difference visible in Fig. 6.22 means that a cut can be applied in the ONLINE processing, requesting an upper limit in the cluster quality. This requirement removes to a large extent clusters that otherwise would spoil the parameters of the straight line fit.

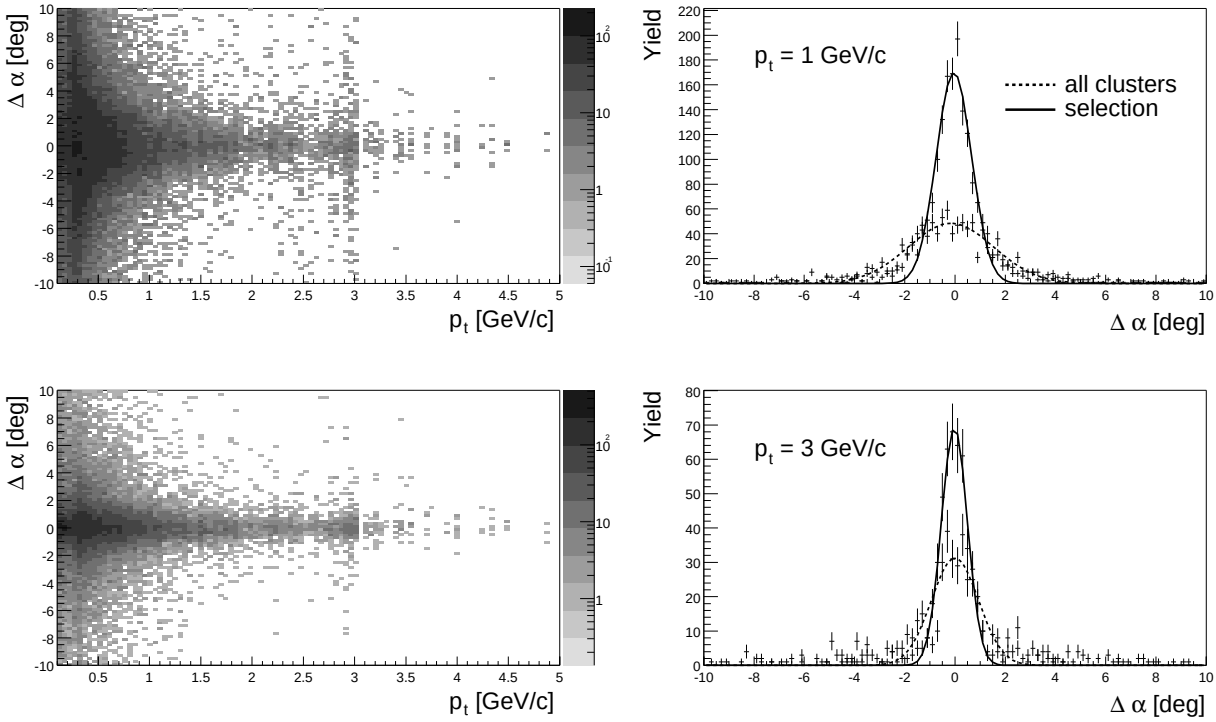

Figure 6.23: Local deflection angle resolution for *tracklets* with (left top) and without (left bottom) cluster quality check as function of the transverse momentum. On the right hand side, projections of the 2-dimensional distributions are show for a transverse momentum of p_t = 1 GeV/c (right top) and p_t = 3 GeV/c (right bottom). Solid (dotted) histograms represent the results with (without) cluster quality selection.

The effect of applying a selection cut for good quality clusters ($A_l \cdot A_r / A_c^2 < 0.0136$) is visualized in Fig. 6.23 where the *tracklet* quality as defined by the deviation of the deflection angle $\Delta\alpha = \alpha_{tracklet} - \alpha_{track}$ is shown with and without the cluster quality selection step for multiplicities corresponding to $< dN/dy >= 8700$. The difference of the expected track deflection to the reconstructed one is plotted versus the transverse momentum of the particle. The figure is done for those *tracklets* that have contributions from only one single particle (called clean *tracklets* in the following). The tails in the distribution are substantially reduced for all transverse momenta. The cluster cleaning improves the local *tracklet* momentum resolution by about a factor of 2 for transverse momenta in the range 1 - 3 GeV/c ,i.e. the RMS width of the $\Delta\alpha$ - distribution changes from 1.7° (0.9°) to 0.7° (0.5°) for 1 GeV/c (3 GeV/c),

respectively. It should be noted that the cluster quality selection is not limited to the simple measure displayed in Fig. 6.22 but can be parameterized in a general way also for non-Gaussian pad response functions in a look-up table. As described in Section 6.3.2 this selection step can be implemented in the *tracklet* reconstruction hardware.

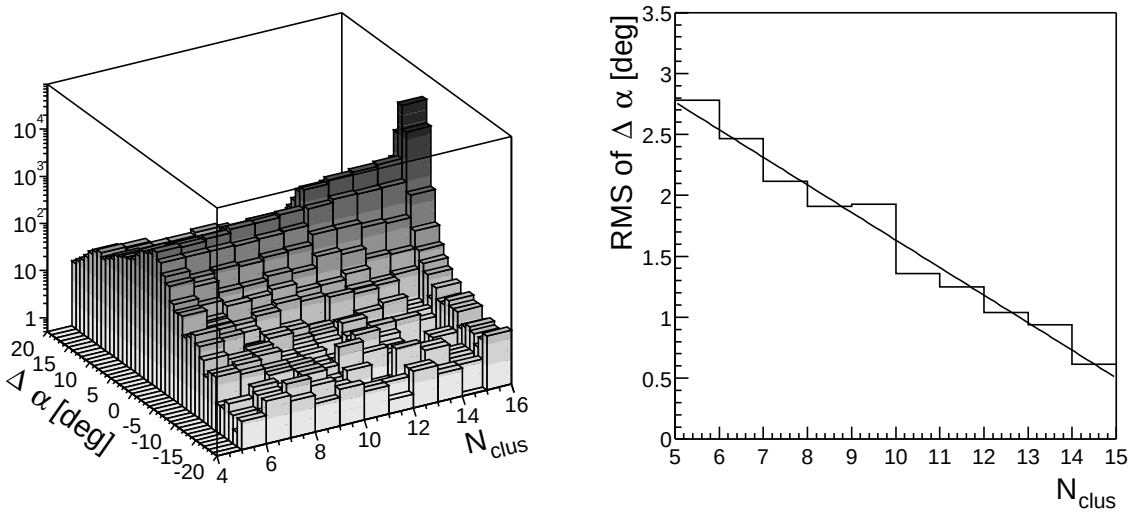

Figure 6.24: Local deflection angle resolution for high p_t clean *tracklets* as function of the number of clusters (left). The right panel shows the RMS width of the difference of the measured deflection angle to the expected angle as function of number of clean clusters contributing to the *tracklet* .

Changes of the *tracklet* resolution with event multiplicity can be expected in terms of a reduced number of clusters that are contributing to a *tracklet* . For a larger multiplicity environment fewer clean clusters passing the cluster quality selection will be found. The effect is shown in Fig. 6.24 where the resolution of the deflection angle with respect to that expected is shown for clean *tracklets* as function of the number of contributing clusters. On the right hand side the corresponding RMS width is given as function of the number of clusters. It is obvious that the width of the deflection is increasing with decreasing cluster number. In order to maintain a sufficient accuracy the number of clusters should not decrease too much. Reasons for losses of clusters within the online algorithm are

- tracks crossing pad rows,
- tracks crossing more than 3 pads in y-direction,
- pile-up.

When a track crosses pad rows (in *z*-direction) the *tracklet* is split into 2 halves. The number of clusters is reduced until 2 *tracklets* are generated in the symmetric case with one half of the original cluster number. As can be seen from Fig. 6.24 the resolution of these *tracklets* is worse by about a factor of 2 - 3 compared to the best cases. This has to be taken into account when trying to match the local *tracklets* in the global tracking step. The quality of the individual *tracklets* is still good enough that no attempt was made to merge them in the LTU. Therefore no communication of processors working on different pad rows is foreseen (see Chapter 5).

6.4.2.4 Local momentum resolution

The position resolution and the distortions due to the TRF and the pile-up define how well a *tracklet* can be measured within a single layer of the TRD detector. Fig. 6.25 presents the deflection angle α

as obtained from a linear regression analysis of the clusters within 3 neighboring pads as described in Section 6.3.2. This selection of clean *tracklets* allows to plot the reconstructed deflection angle α versus the momentum of the particle that generated the tracklet. Note that the momentum is taken from the production vertex of each particle.

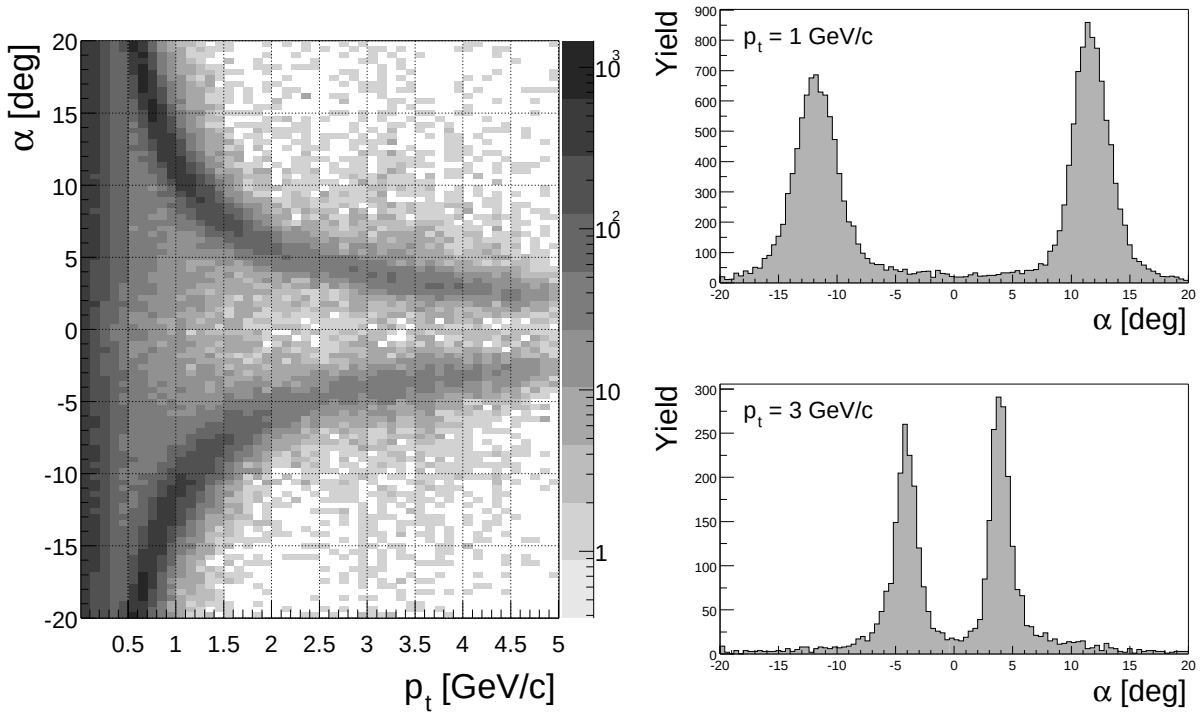

Figure 6.25: Tracklet deflection angle for clean tracklets as function of the transverse momentum (left). On the right hand side two projections are shown for transverse momenta of p_t = 1 (top) and 3 (bottom) GeV/c .

Fig. 6.25 makes use of the tail cancellation method described in Section 6.4.2.1 and the cluster quality selection described in Section 6.4.2.3. Under those conditions the position resolution for clean *tracklets* is clearly sufficient to select tracks of about 3 GeV/c with good efficiency and high discrimination power against low momentum *tracklets* , e.g. by requesting a deflection angle of $\alpha < 6^\circ$.

It is, however, apparent from Fig. 6.25 that, besides background due to energy loss and elastic scattering that widen the branches around the expected deflection angles, there is a strong component originating from low momentum particles. Low momentum particles originating from conversions and interactions are producing *tracklets* with an almost uniform deflection angle distribution.

With the cluster quality selection defined above the local transverse momentum resolution of the online method is shown in Fig. 6.26. Large differences are observed between the different scenarios. The resolution obtained with the standard scenario is worse by about a factor of 2 - 3 with respect to the academic case, where the TRF is modeled by a δ-function. Applying a digital filter recovers the resolution at low momenta and substantially improves the situation for high transverse momenta. For transverse momenta of p_t = 3 GeV/c a resolution of $\sigma(p_t)/p_t$= 20% is obtained.

6.4.2.5 Local Tracking Efficiency

The efficiency of the trigger system can be decomposed into two steps, the track finding efficiency and the track selection efficiency. Track finding is achieved by asking for a certain number of clusters contributing within three neighboring pads. Track selection requires a certain number and quality of contributing clusters resulting in a sufficient quality of the fit parameter.

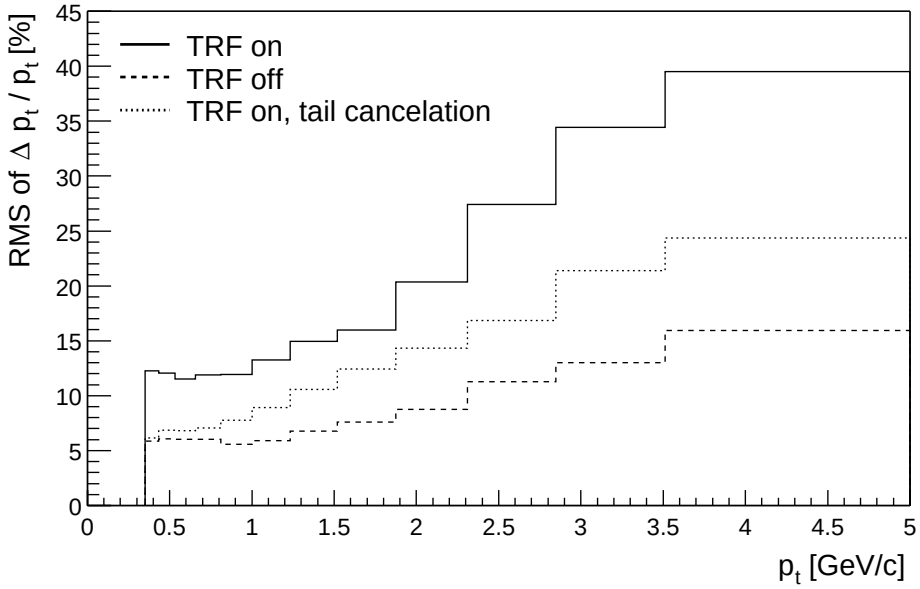

Figure 6.26: Local momentum resolution for clean *tracklets* as a function of the primary transverse momentum for an event multiplicity of $< dN/dy >$=8700. The different line styles correspond to different scenarios: TRF on (solid), TRF off (dashed) and employing a digital tail cancellation (dotted).

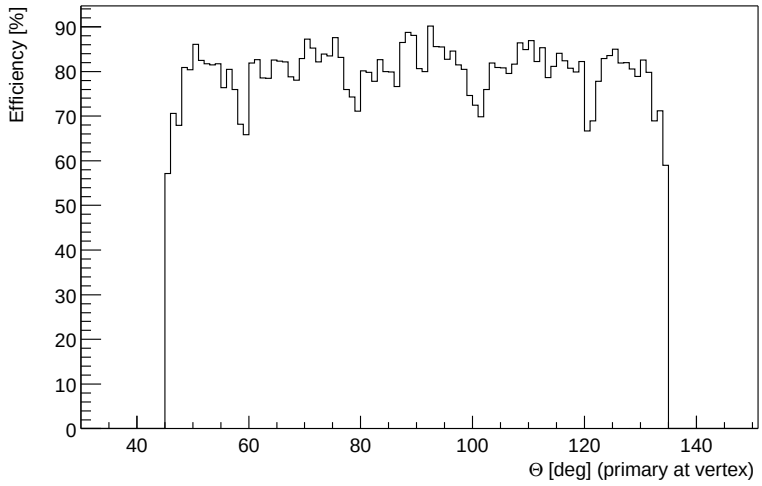

Figure 6.27: Tracklet efficiency for stiff tracks (p_t>1GeV/c) as a function of the polar angle. The minimum number of clusters required for a *tracklet* is 5.

Polar angle dependence of local track finding efficiency

As indicated above, due to the chamber and pad geometry and the range in track topologies the tracks will not always stay within one pad row. Especially at forward/backward angles, a substantial number will be split into two pieces. It was analyzed to which extent the efficiency of the *tracklet* finding algorithm suffers due to this. The average number of clusters per *tracklet* as function of the polar angle is found to be constant, and the variation of the *tracklet* length with polar angle is rather weak. The dependence of the track finding efficiency (displayed in Fig. 6.27) is even weaker since *tracklets* are accepted if the contributing number of cluster is above a certain minimum value ($N^{min}_{cluster}$=5 in the

specific case). The polar angle coverage of the online tracking scheme is rather uniform except for the dips visible in Fig. 6.27 at polar angles of 60° , 80° , 100° and 120° , corresponding to the chamber boundaries.

Local selection efficiency

The influence of selecting *tracklets* with deflections corresponding to different momenta is presented in Fig. 6.28. The efficiency is defined as number of *tracklets* originating from particles from the primary vertex that were found with a deflection angle smaller than the cut value, normalized to the total number of primary particles with according momenta emitted into the solid angle of the TRD. The efficiency is evaluated as function of the transverse momentum taken from the primary vertex (referred to as primary transverse momentum in the following).

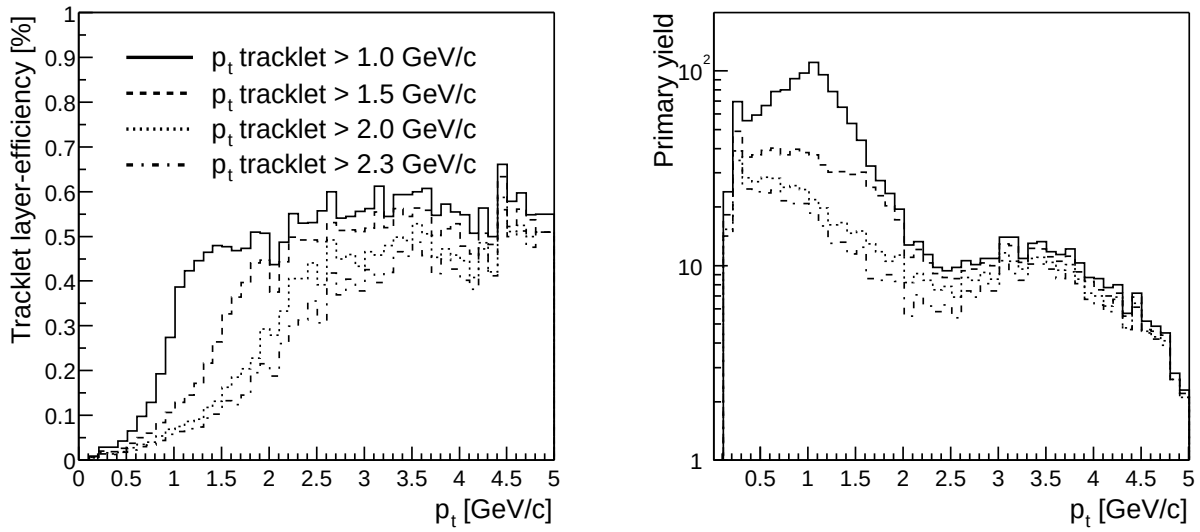

Figure 6.28: Online *tracklet* selection efficiency as function of transverse momentum. The histograms show the efficiency (left panel) and the total accepted number of primary *tracklets* (right panel) under various deflection cuts corresponding to transverse momenta of 1, 1.5, 2 and 2.3 GeV/c for full multiplicity events at a magnetic field of B = 0.4 T.

The efficiency curves show a typical threshold behavior. By decreasing the cut value (i.e. increasing the lower momentum limit) the threshold is smeared out and the plateau efficiency at large momenta decreases slightly. The effect of the selection on the measured spectra is shown in the right panel of Fig. 6.28. A substantial decrease in total number of *tracklets* is achieved by employing a larger transverse momentum threshold. Note that the peak in the spectra at $p_t \approx$ 3.5 GeV/c stems from the additional electrons of the input distribution. The threshold needs to be optimized since there is a balance between the total number dominated by low transverse momentum *tracklets* that should be minimized and the local tracking efficiency at large transverse momentum that should be maximized. The total number of found *tracklets* should be minimized since accepting more low momentum *tracklets* will increase the occupancy at the global matching stage and present higher demands to the readout bandwidth of the trigger system.

Quality dependence of local tracking efficiency

Another effect of the cluster quality and the resulting *tracklet* quality is shown in Fig. 6.29 where the influence of the TRF is demonstrated. With the better cluster quality available by the undistorted signal

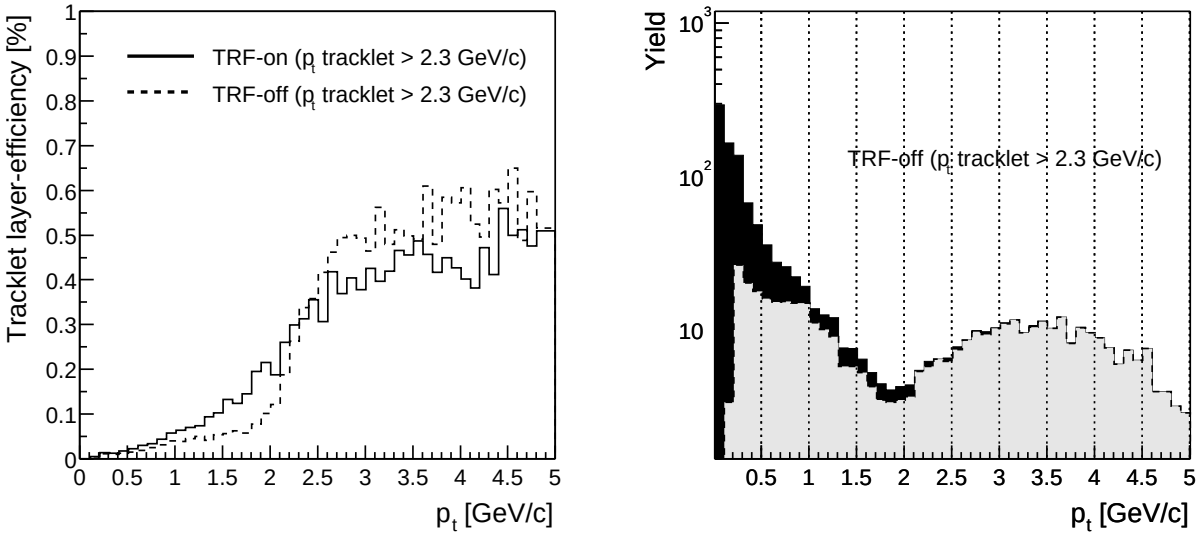

Figure 6.29: Same as Fig. 6.28 with a selection cut of 2.3 GeV/c with and without the time response function (left). The right panel shows the spectral shape of the selected tracklets: the black histogram corresponds to the total numbers, of which the grey portion represents the contribution of tracklets originating from tracks from the primary vertex.

shape, the threshold is much sharper and the yield of low momentum particles below the threshold of p_t= 2 GeV/c is reduced by about a factor of 1.6. In both scenarios a plateau in the efficiency is reached at the target momentum of p_t= 3 GeV/c . The plateau values of the efficiency differ by about 5%. On the right panel of Fig. 6.29 the spectral shape of the selected tracklets is shown for the 'TRF off' case. It is evident that even under the favorable conditions of good resolution, the total number of found tracklets is dominated by tracklets originating from low momentum particles. Although the overall number of found tracks does not differ significantly from the case with the time response function turned on, the better definition of the high momentum threshold for true tracks from the primary vertex is again an argument in favor of applying deconvolution schemes to the pulse shapes trying to remove the influence of the tail.

6.4.2.6 Multiplicities from local tracking units

The load of the TRD chambers in terms of found *tracklets* is studied in Fig. 6.30. This is relevant in order to determine the necessary bandwidth within the trigger system from the local processors to the global tracking/matching unit. The largest local threshold momentum, for which the plateau efficiency is reached at p_t = 3 GeV/c , was determined to be $p_t^{threshold}$=2.3 GeV/c as demonstrated in Figs.6.28 and 6.29. The number of *tracklets* found per readout chamber with a cut corresponding to p_t = 2.3 GeV/c is shown in the left panel for different event multiplicities. The distributions are Gaussian shaped and can be quantified by a mean and a width. In order to define the requirement of the transmission bandwidth, the mean and the maximum number of *tracklets* per chamber are plotted on the right panel of the figure as function of the event multiplicity.

Since the shipping of the data from all the different TRD chambers to the GTU is done in parallel, the chamber with the largest number of *tracklets* determines the dead time. Therefore a minimal requirement for the design of the readout scheme can be derived: the trigger hardware has to have a bandwidth for about 40 *tracklet* per chamber (maximum value for a full multiplicity event with a selection threshold of 2 GeV/c).

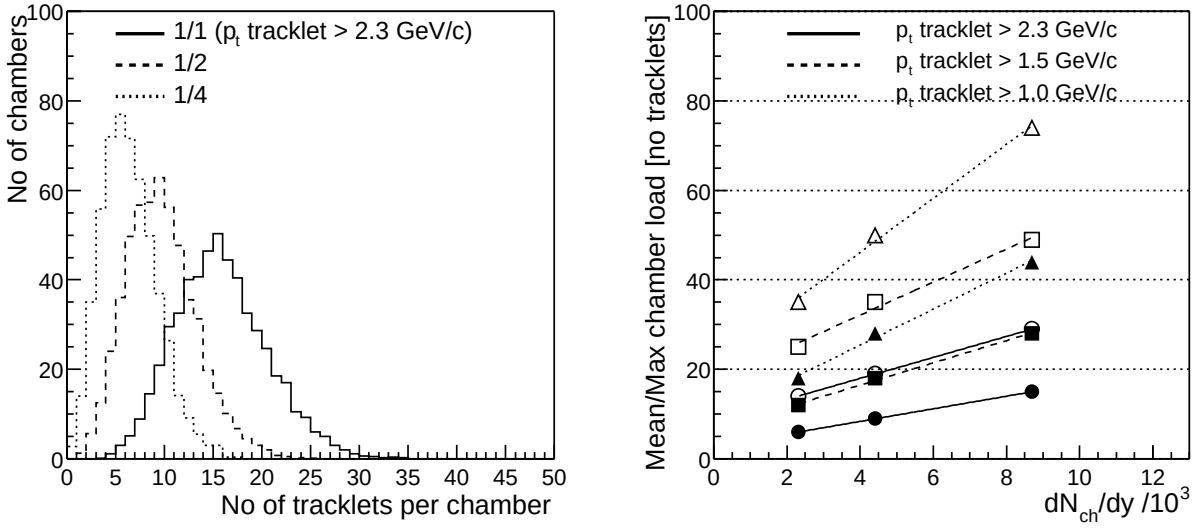

Figure 6.30: Chamber load. The distribution of the number of found *tracklets* per readout chamber for various event multiplicities is shown in the left panel. The multiplicity dependence of the mean (solid symbols) and the maximum (mean+3σ level, open symbols) *tracklet* number is given in the right diagram for various selection thresholds.

6.4.3 Tracklet Matching - Global Tracking Unit

All track candidates of the individual layers consisting of position, angle, amplitude, quality etc., are projected to a (virtual) middle plane by computing the proper azimuthal angle of intercept, the longitudinal *z* - position and the deflection angle α. This transformation can be done already within the LTU, practically without any additional time. The principal concept of the global tracking units is sketched in Fig. 6.31.

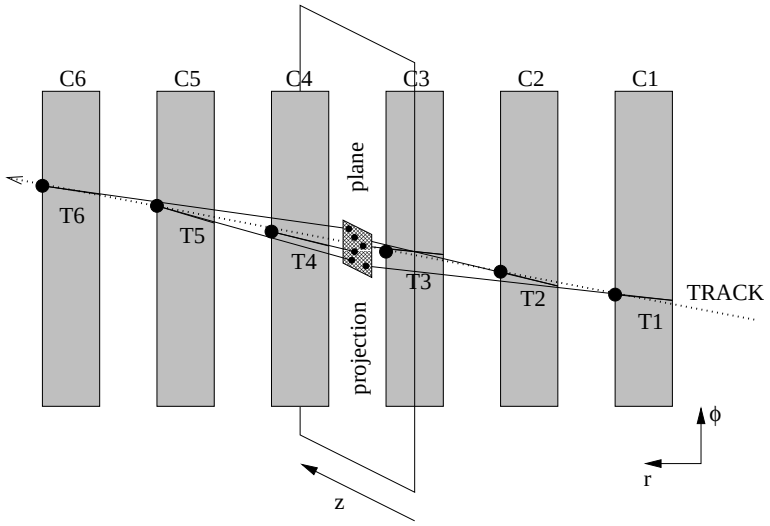

Figure 6.31: Global tracking concept

The basic task is to count the number of *tracklets* within certain regions. This can be achieved most conveniently by a global histogram, where the three matching quantities (azimuthal angle φ, *z*-coordinate and deflection angle α) are recorded. The granularity of the global matching histogram needs to be adjusted to the extrapolation uncertainty caused by the resolution of the *tracklet* parameters obtained and the occupancy of the global histogram. For a good performance a high p_t-cut from the single layers helps to reduce the occupancy. As shown in Fig. 6.30 under the condition of a relatively high local momentum cut of 2 GeV/c on average 20 *tracklets* are shipped to the GTU, i.e. the histogram has 8100 entries per central event.

To achieve a sufficient global tracking efficiency and a sufficient electron-pion separation 3 out of possibly 6 *tracklets* are required for the definition of a good primary track. With a single layer tracking efficiency in the order of $\varepsilon(p_t > 3\text{GeV}/c) \approx 0.5$, the expected global tracking efficiency can be evaluated to

$$p_{global} = \sum_{i=3}^{6} \frac{6!}{i!(6-i)!} p_{local}^{i} (1 - p_{local})^{(6-i)} = 0.66$$

provided that all the inspection widths of the global histogram extend to 3σ in the φ, *z* and α direction. To achieve a global tracking efficiency of 90% local tracking efficiencies in the order of 67% would have to be realized.

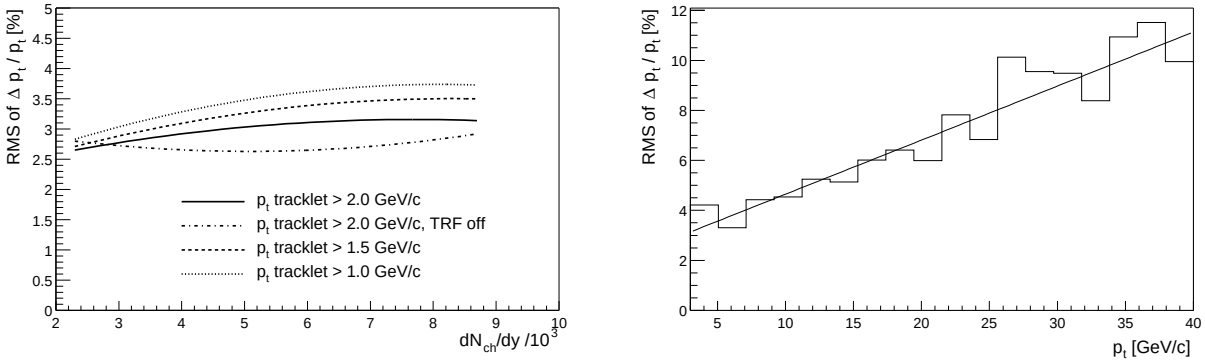

Figure 6.32: Momentum resolution of online globally reconstructed tracks. The left panel displays the global relative momentum resolution $\sigma(p_t)/p_t$ for tracks with primary transverse momenta of p_t = 3 GeV/c as function of multiplicity for various local transverse momentum selection thresholds. On the right panel the transverse momentum dependence of the global momentum resolution is shown for full multiplicity conditions of $< dN/dy >$=8700.

The important role of the GTU is i) to define a sharp transverse momentum threshold close to 3 GeV/c , ii) to select stiff tracks of electrons (positrons) and iii) to calculate invariant masses or find jets. The global transverse momentum is calculated with the same formalism like the local transverse momentum by evaluating the global deflection angle with respect to the nominal primary vertex direction. The global momentum resolution as function of p_t is shown in Fig. 6.32. With a common set of parameters like in Table 6.6 the momentum resolution for tracks originating from the primary vertex is found to be rather independent of the event multiplicity (occupancy) and the local momentum selection threshold. Typical values are $\sigma(p_t)/p_t$ = 3 - 4 % for transverse momenta of p_t= 3 - 4 GeV/c . Due to the longer lever arm for determining the transverse momentum, this resolution does not depend significantly on the local resolution (as shown by the "TRF-off" calculation in Fig. 6.32). With matching widths as chosen in Table 6.6 the distortions due to chance coincidences in the GTU matching histogram are rather weak even under the worst case condition of the full multiplicity of $< dN/dy >$=8700. In summary, keeping a distance of 3σ to the target transverse momentum a threshold of $p_t > 2.7$ GeV/c can be implemented in order to select primary tracks with a transverse momentum of $p_t >$ 3 GeV/c . Note that electrons experience significant momentum losses due to bremsstrahlung, i.e. the expected efficiency

losses are larger than for Gaussian distributions (see Fig. 6.33).

The momentum resolution shown in Fig. 6.32 stays below 5% for transverse momenta up to 10 GeV/c . This resolution is obtained for all charged particles. Therefore one has a large flexibility of defining triggers requiring a number of stiff tracks in a given solid angle (jet triggers), e.g. it is certainly possible to select three stiff tracks within the solid angle of one readout chamber requiring for each of the tracks a transverse momentum of p_{t}> 5 GeV/c .

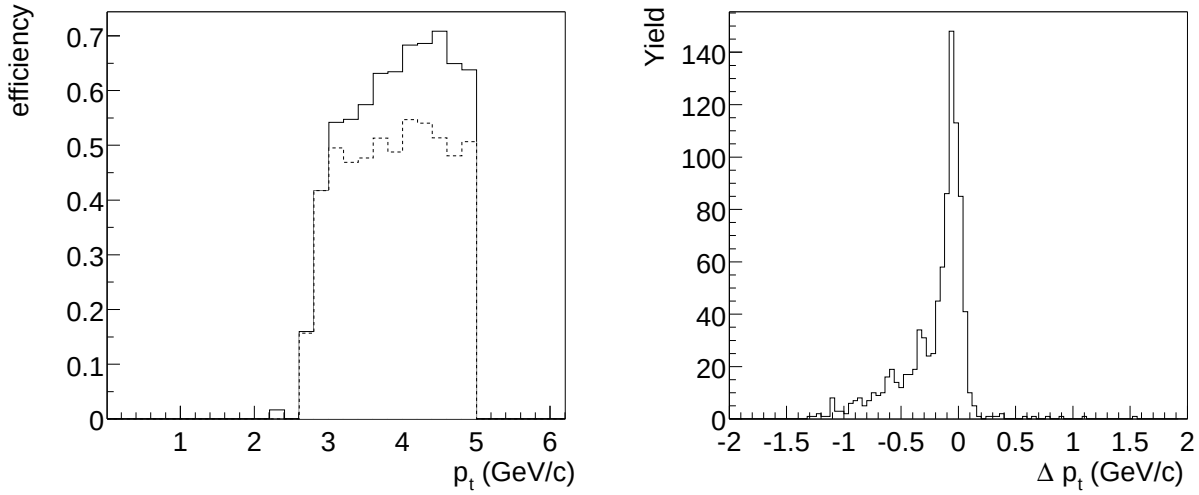

Figure 6.33: GTU tracking efficiency for electrons in a low multiplicity (400) environment (left). The solid histogram represents the track finding efficiency, the dashed histogram displays the efficiency to reconstruct the primary electron with a transverse momentum that deviates less than 10% from the primary transverse momentum. The right panel shows the momentum loss distribution of electrons when entering the first plane of the TRD.

6.4.4 GTU tracking efficiency

The efficiency of the GTU was studied in comparison to the offline performance and under various scenarios. The detection efficiency as function of momentum for clean electron tracks (at event multiplicity 400) is shown in Fig. 6.33. A particle is called detected, once its reconstructed momentum has passed a certain threshold ($p_t^{rec} > 2.7\mathrm{GeV}/c$). In the figure the number of those particles is plotted versus their primary momentum normalized to all the primary particles that are emitted into the solid angle of the TRD detector ($45^\circ < \theta_{lab} < 135^\circ$). The global track finding efficiency (solid histogram in Fig. 6.33) is rising as function of the primary transverse momentum due to the long radiation tail of the incident electron momentum distribution. This is evident when inspecting the momentum loss distribution $\Delta p_t = p_t^{incident} - p_t^{primary}$ that is plotted on the right hand side of Fig. 6.33 for electrons that are emitted with a transverse momentum of 3 GeV/c from the primary vertex. Cutting this distribution at $\Delta p_t = -0.3\mathrm{GeV}/c$ corresponding to a global trigger selection threshold of p_{t}^{min} = 2.7 GeV/c reduces the number of found tracks by 28%. At the threshold the momenta of the found tracks are essentially all within a 10% window around the original momentum from the primary vertex, as shown by the dashed histogram in Fig. 6.33. This fraction of about 50% of all the emitted electron that are found within a 10% window stays constant with increasing transverse momentum while the track finding efficiency rises due to the constant selection threshold. The correct momentum of those additional tracks that have experienced a substantial energy loss before reaching the TRD can only be determined by tracking through the full ALICE detector and identifying the initial curvature of the track with the ITS and TPC.

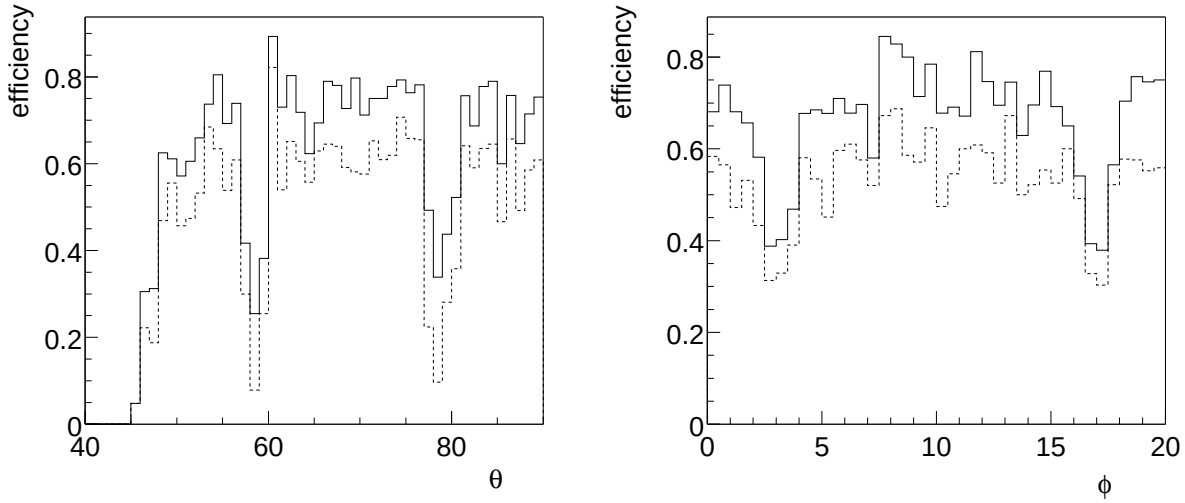

Figure 6.34: GTU tracking efficiency for electrons at low multiplicity versus polar angle (left) and versus projected azimuthal angle (right). The line types of the histograms are the same as in Fig. 6.33 (left panel).

The uniformity of the efficiency over the solid angle is shown in Fig. 6.34, demonstrating the validity of the underlying tracking concept. The values are obtained by averaging over the signal part of the transverse momentum spectra (Fig. 6.17) with the efficiencies shown in Fig. 6.33. The dips visible in the efficiency are caused by the structural material and the gaps in the acceptance. For the azimuthal distribution the whole angular acceptance was projected onto the opening angle of a single chamber ($\Delta\varphi = 20^\circ$). Since primary emission angles of the tracks are displayed in the histogram, the inefficiency caused by the frames and gaps are shifted by the deflection angle of those tracks, e.g. $\alpha \approx \pm\ 3^\circ$ for the electron/positron momenta under study here.

The summary of the results obtained with the GTU tracking algorithm are shown in Fig. 6.35 as function of the normalized event multiplicity. The normalization is such that unity corresponds to $dN/dy = 8700$. The left panel shows the track finding efficiency for primary electrons with transverse momenta of more than 3 GeV/c including the requirement $p_{\mathrm{t}}^{rec} > 3$ GeV/c . The number is calculated by integrating over the signal electron spectrum as shown in Fig. 6.17. Figure 6.35 (middle) shows the total number of reconstructed tracks originating from any charged particle and Fig. 6.35 (right) shows the number of found electron background tracks, i.e. tracks that originate from secondary electrons that will not be distinguishable from primary electrons due to the TR signature.

A systematic study was performed aiming at the reduction of secondary electron tracks to the level of 1-2 for full multiplicity events for trigger rate reasons (see section 6.5). This can be achieved by cutting on the quantities listed in Table 6.6. In order to suppress secondary electron tracks that are mostly due to conversions of γ-rays in the inner parts of the ALICE detector as much as possible in addition to *tracklet* quality cuts the curvature of the track was determined without reference to the primary vertex. This can be done for track candidates with 3 or more contributing *tracklets* in different planes. The derived quantity, the unconstrained transverse momentum (called p_{t}^{free}), has worse momentum resolution but was found to efficiently suppress background from secondary electrons. The effect is visible in Fig. 6.35 (right) when comparing the solid squares and triangles corresponding to Cut A and Cut B as given in Table 6.6.

In order to map out the dependencies of the efficiency and the background, several scenarios are investigated in Fig. 6.35:

1. Offline tracking performance with default digitization scenario (solid circles),

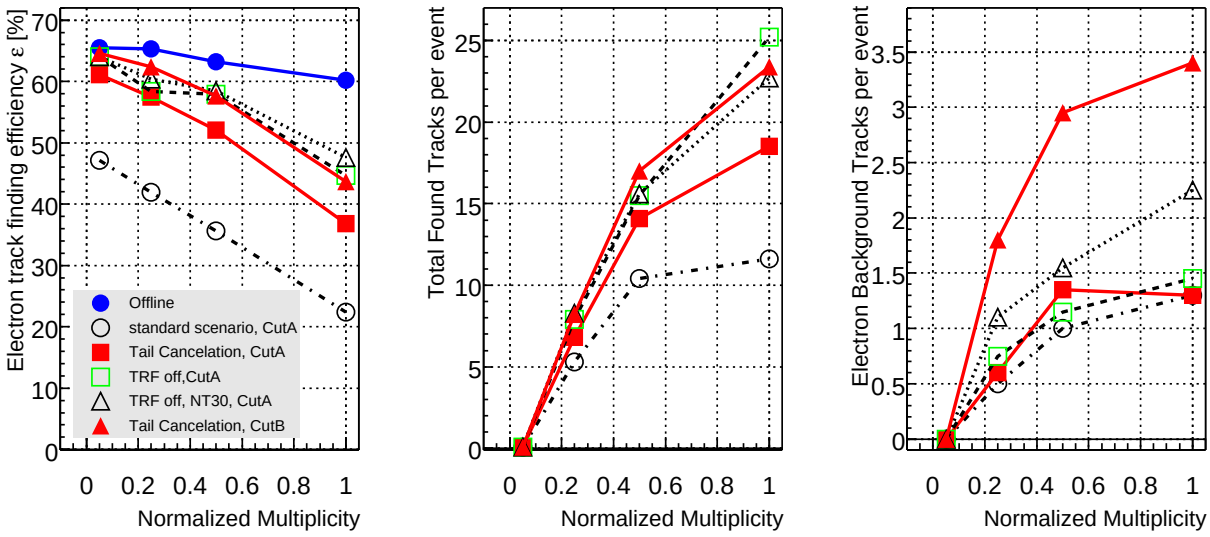

Figure 6.35: GTU tracking performance as function of normalized event multiplicity. The normalization is such that unity corresponds to $dN/dy = 8700$. The left panel shows the track finding efficiency for primary electrons with transverse momenta of more than 3 GeV/c . Figure 6.35 (middle) shows the total number of reconstructed tracks originating from any charged particle and Fig. 6.35 (right) shows the number of found electron background tracks.

Table 6.6: Global tracking parameters.

quantity	Cut A	Cut B
matching width in azimuthal angle	0.11°	0.13°
matching width in z-direction	8 cm	8 cm
matching window for deflection angle	2.4°	2.4°
local momentum threshold	2.3 GeV/c	2.3 GeV/c
minimum number of clusters per *tracklet*	6	5
upper χ^2-limit for accepted *tracklets*	4.0	4.0
minimal number of *tracklets*	3	3
minimal number of planes crossed by track	4	4
threshold for p_t^{rec} - p_t^{free}	1.7 GeV/c	2.5 GeV/c
global momentum threshold	2.7 GeV/c	2.7 GeV/c

2. Standard scenario: Online tracking scheme based on 15 time bins with the nominal TRF (see Fig. 11.8) (open circles),
3. Same as 2 but including tail cancellation (see Section 6.4.2.1) (solid squares),
4. Same as 2 but with a δ-function as TRF (open squares),
5. Same as 4 but with 30 time bins (open triangles).

The filled triangles in Fig. 6.35 correspond to scenario 3 with different parameters (Cut B from Table 6.6). Several observation are made from Fig. 6.35:

- The track finding efficiency is at best about 65% for low multiplicities as determined with the

full offline tracking algorithm (solid circles). This value is caused by the radiation losses of the electrons in the material inside of the TRD and the application of a fixed momentum threshold.

- The standard online tracking scenario (open circles) reaches for low multiplicities a peak efficiency of about 50% only. The reduction with respect to the offline case is due to i) implementing the background suppression cuts (Cut A of Table 6.6) and ii) distortions of the *tracklet* parameter due to the TRF. The TRF is responsible for an efficiency loss of about 12% of the total number of electrons as visible in Fig. 6.35 when replacing it by a δ-function (open squares). The various selection cuts cost in total about 4% of the electrons (for low multiplicities).

- 80%-90% of the losses caused by the TRF can be recovered by implementing the digital tail cancellation described in Section 6.4.2.1 (filled squares) employing a single pole/zero network. Since this feature turns out to be useful for the online tracking efficiency, it is planned to implement the digital filter into the digital chip. This scenario is considered the effective default for the online tracking.

- The track finding efficiency decreases linearly with increasing multiplicity. The decrease of the efficiency is proportional to the occupancy (see Fig. 6.21). Such a scaling is to be expected due to the method to remove overlapping clusters while in the offline case deconvolution algorithms can be applied.

- For full multiplicity conditions increasing the number of independent time samples in the TRF off case (open triangles) improves the efficiency by about 10%. To which extent this observation holds with a tail cancellation algorithm and realistic pulse shapes needs to be studied further. This scenario would suggests to optimize the total drift time under the constraint of a TRF determined by the drifting ions.

- The improvement in the electron efficiency is reflected in the average number of tracks that are found under the global momentum cut (middle panel).

- The number of background electron tracks (right panel) depends on the set of cut values. The most crucial cut for removing secondary electrons is constructed from the unconstrained transverse momentum fit. The effect is seen by comparing Cut A (solid squares) to Cut B (solid triangles). At normalized multiplicities of 0.25 and 0.5 relaxing the selection condition increases the electron finding efficiency by about 6 - 8%. This is, however, accompanied by an increase in the background yield by about a factor of 2 - 3.

With the complex features observed for the online tracking algorithm and the large uncertainties of the input distributions it is difficult to make quantitative statements that are generally valid. It should be stressed, however, that the event multiplicities and the spectral shapes of the hadrons used in the simulation represent the worst case scenario. Even under those conditions the electron trigger scheme described above allows to reconstruct a sufficient number of resonances as will be shown in Section 6.5.

6.4.5 Electron identification

In addition to the track finding problem described in the previous section, the found tracks need to be identified as electrons for the e^+e^-trigger. For a jet trigger application this step is omitted. The left panel of Fig. 6.36 presents the pulse heights distribution in a single chamber obtained for pions and electrons with transverse momenta of $3 < p_t < 5$ GeV/c when integrating over the full drift time of the readout chamber in low multiplicity events. A threshold can be set so that by keeping 90% of the electrons we reject the pions with a factor of 25.

A more sophisticated analysis can be done following the procedure described in Section 11.5. The likelihood shown in the right panel of Fig. 6.36 is calculated from the normalized distributions of the

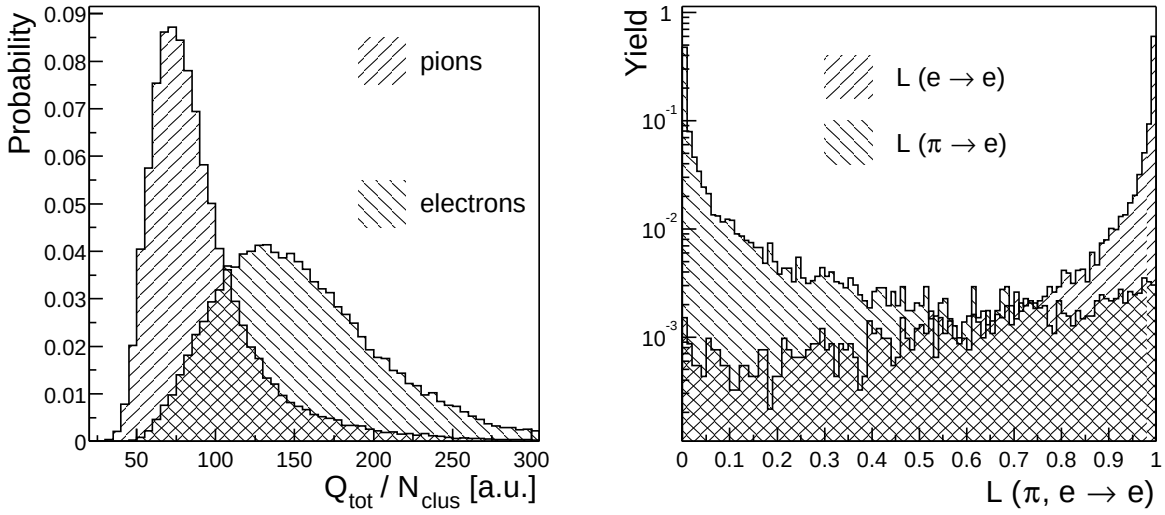

Figure 6.36: Pulse height distributions for electrons and pions with transverse momenta $p_t > 3$ GeV/c integrating all time bins in one chamber (left) and derived likelihood distributions to infer an electron from the electron ($L(e \to e)$) and pion ($L(\pi \to e)$) pulse height distributions (right).

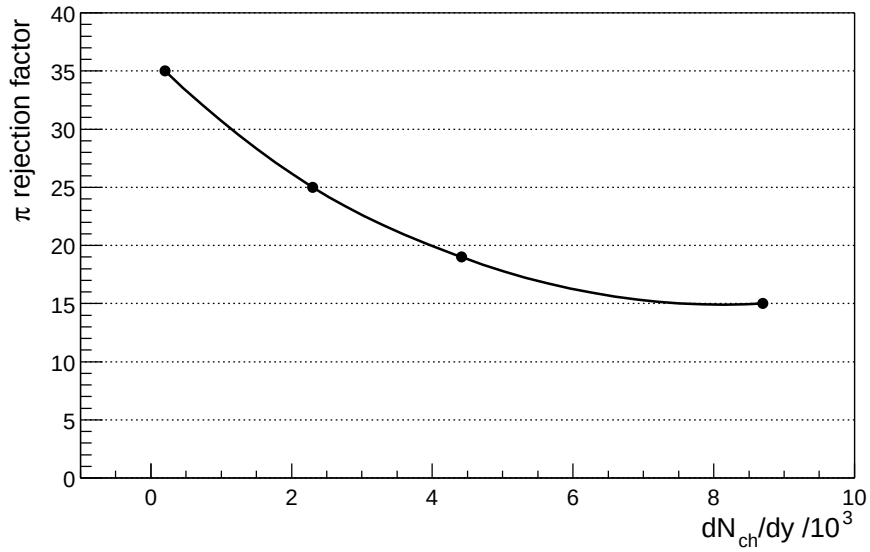

Figure 6.37: Pion rejection factor for 90% electron efficiency, determined from the average charge in half chamber, as function of the multiplicity.

energy deposited in one TRD module by electrons and pions, which can be implemented as two look-up-tables. For a given globally reconstructed track the information consists of n values of normalized energy deposit ΔE_i for a track reconstructed from n tracklets. From the LUTs the probabilities p_i^e and p_i^π are obtained, thus the total probabilities can be calculated as $P_e = \prod p_i^e$ and $P_\pi = \prod p_i^\pi$ over the n layers. The likelihood for the particle to be an electron is given by the ratio $P_e/(P_e + P_\pi)$. To this value a cut is applied, chosen in such a way that 90% of the electrons are retained. The result of the online PID procedure is plotted in Fig. 6.37 as function of the centrality of the collision. The drop in the rejection power is caused by the decreasing number of tracklets that form the globally accepted track.

6.4.6 Background

Background sources for high p_t electrons:

- background electrons: conversions, δ - electrons,
- fake tracks,
- misidentified charged pions.

While the fake tracks and secondary electrons are a problem for all triggers, the background due to charged pions affect only the di-electron trigger. As visible by the solid squares of Fig. 6.35 (middle), 18 tracks are found on average per central collision, of which 1.4 are clean electron tracks (right panel of Fig. 6.35). Applying the online particle identification capability described in Section 6.4.5 reduces the totally found tracks to about one electron track candidate per central collision. The background due to misidentified pions and fakes and the one from secondary electrons are, for restrictive parameters like 'Cut A', of the same order of magnitude.

The background multiplicity is of crucial importance for the fake trigger rate that is discussed in Section 6.5.3. It should be noted that the background multiplicity can be reduced further in a post-processing of the event in the HLT compute farm:

- The pion suppression of misidentified pion tracks can be improved to the offline performance (see Section 11.5).
- The secondary electron tracks passing through the selection cuts have a wide true momentum distribution. 60% of them have transverse momenta below 3 GeV/c . Tracking of the TRD electron candidates through the TPC would determine the curvature of the track with a better accuracy and can be used to remove this component. The large momentum background tracks result from photon conversions in the inner part of the ALICE detector. They can be recognized at least partially by full tracking through the TPC and ITS as well.

The corresponding trigger rates for e^+e^-coincidences and the important role of the HLT is discussed in Section 6.5.3.

6.5 Performance

The various physics observables are discussed in [3]. A typical example of an interesting hard probe physics observable is the family of Υ resonances [9]. The performance of the trigger scheme described above is presented in this section for the Υ system.

6.5.1 Generalities of Pb-Pb collisions

With an extrapolated inelastic nucleon-nucleon cross section of $\sigma_{NN} = 60$ mb a total reaction cross section of $\sigma_{PbPb} = 7.5$ b is anticipated. At a luminosity of $\mathcal{L} = 1 \cdot 10^{27}\text{cm}^{-2}\text{s}^{-1}$ this cross section gives rise to minimum bias interaction rate of $R_{int}^{MB} = 7.5$ kHz. In order to make use of the high quality tracking available within the ALICE TPC, only the fraction of events without pile-up within the TPC drift time of $T_{drift}^{TPC} = 88\mu$s is considered at this stage for further analysis. Therefore at the trigger level the rate is reduced to a past protected interaction rate of $R_{past}^{MB} = 3.9$ kHz, while for high level trigger processing, a rate of $R_{past\&future}^{MB} = 2.0$ kHz is anticipated. The numbers are summarized in Table 6.7 for luminosities from $0.5 - 1.0 \cdot 10^{27}\text{cm}^{-2}\text{s}^{-1}$.

Table 6.7: Event and trigger rates for Pb+Pb at 5.5 A TeV.

$\mathcal{L}(\mathrm{cm}^{-2}\mathrm{s}^{-1})$	R_{int}^{MB} (kHz)	R_{past}^{MB} (kHz)	$R_{past\&future}^{MB}$ (kHz)
$1\cdot 10^{27}$	7.5	3.9	2.00
$7.5\cdot 10^{26}$	5.7	3.5	2.10
$5\cdot 10^{26}$	3.8	2.7	1.95

With the past-future protection, the optimum conditions are reached at a luminosity of $\mathcal{L} = 7.5\cdot 10^{26}\mathrm{cm}^{-2}\mathrm{s}^{-1}$. For the following estimates an integrated luminosity of $\mathcal{L}_{int}^{PbPb} = 0.5/\mathrm{nb}$ per year is assumed, based on a time averaged luminosity of $\mathcal{L} = 5\cdot 10^{26} cm^{-2}s^{-1}$ and an effective data taking time of 10^6 s within one month of heavy ion running per year [10]. As can bee seen from Table 6.7, due to the past/future protection imposed by the TPC drift time, the number of acceptable events does not change when the luminosity is increased by a factor of 2.

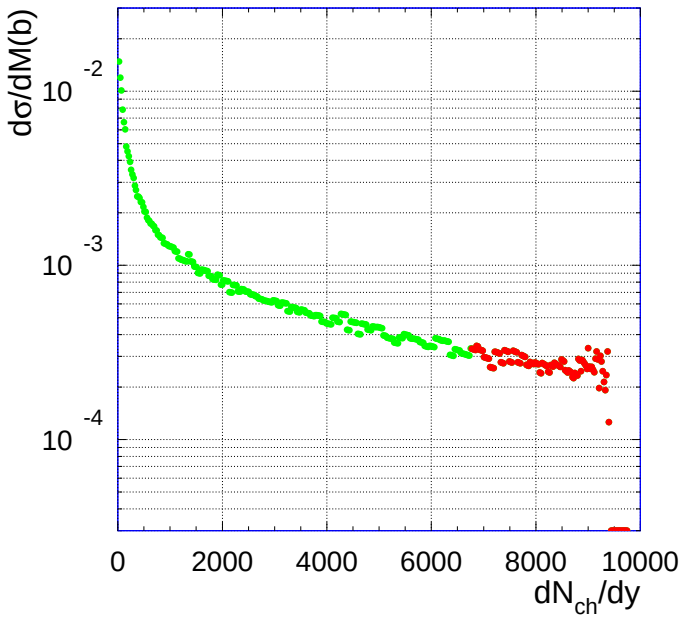

Figure 6.38: Differential cross section versus multiplicity for Pb+Pb at 5.5 A TeV.

The distribution of multiplicities can be obtained by assuming that the multiplicity scales with the number of participants that are calculated by the nuclear overlap integral as function of the impact parameter. Fig. 6.38 shows the scenario with a mean value of the rapidity density $< dN/dy >$=8000 for the most central 10% of all the reactions. The event class of the most central 10% of the cross section is called in the following *CEN*. It starts at a multiplicity of $dN/dy_{low} = 6770$ and is marked in Fig. 6.38 by the dark circles.

Hard probes are produced according to the number of primary NN - collisions that can be obtained from the same formalism. Background is created according to the occupancy of the TRD chambers that scales linearly with the multiplicity (see Fig. 6.21).

6.5.2 Υ count rates

The integrated production cross section of the Υ resonances at $\sqrt{s} = 5.5$ TeV is obtained by extrapolating existing data in the framework of the color evaporation model [11–13]. For Pb-Pb collisions a total production cross section for the combined Υ states (Υ, Υ' and Υ'') decaying into two electrons (B=2.52%) of $B\sigma^{\Upsilon}_{PbPb} = 570$ μb is predicted [12]. The relative fractions are Υ: Υ' : Υ'' = 1 : 0.3 : 0.1. The cross section estimate is based on an extrapolated pp cross section of $d\sigma_{pp}/dy|_{y=0} = 3$ nb. The corresponding rapidity distribution is approximately flat over ± 4 units of rapidity resulting in an integrated cross section of $\sigma_{pp} = 22.5$ nb. Scaling from pp to AA collisions is done by the scaling law observed in the Fermilab fixed target data

$$\sigma_{AA} = A^{2\alpha}\sigma_{NN},$$

with $\alpha = 0.95$ [14] .

With these assumptions a minimum bias rapidity density for $\Upsilon \rightarrow e^+e^-$ of $dN/dy|_{y=0} = 1 \cdot 10^{-5}$ is predicted for minimum bias Pb-Pb collisions. An integral number of $N^{\Upsilon}_{int} = 2.85 \cdot 10^5$ is produced within one year of ALICE running. Note that this number is obtained from scaling pp results with the number of primary collisions, i.e. no suppression or enhancement from the QGP is considered at this level. The current estimate is lower by a factor of 1.6 as compared to the numbers used in the Technical Proposal [3] due to the use of the empirical mass scaling. With all the extrapolations and uncertainties the following numbers can only be considered a case study that describe the capability of the TRD trigger system.

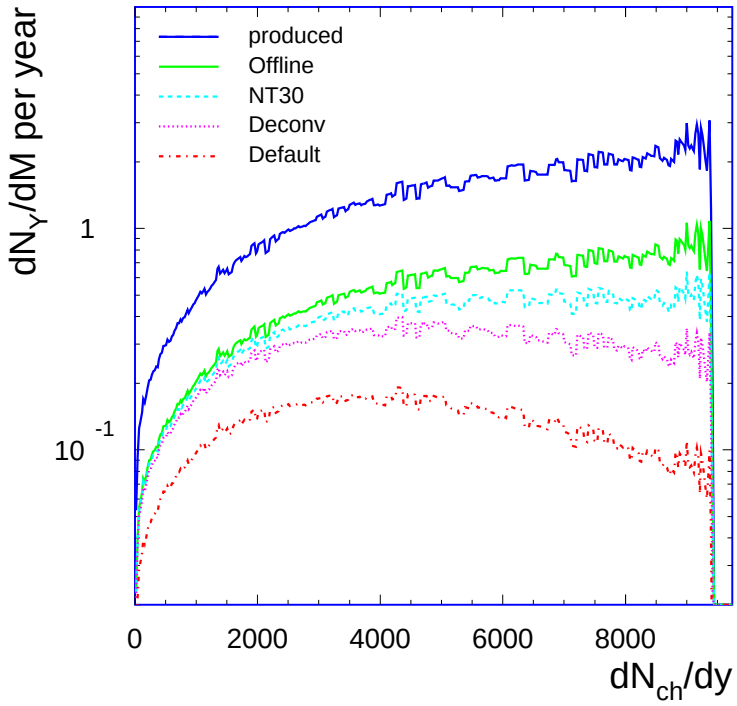

Figure 6.39: Differential Υ- yields for various scenarios as function of multiplicity for Pb+Pb at 5.5 A TeV. The scenarios Offline (solid), NT30 (dashed), Deconv (dotted) and Default (dashed-dotted) correspond to the scenarios 1,5,3, and 2 of Section 6.4.4.

As shown in the Technical Proposal [3] the geometrical acceptance ε_{geo} for detecting an electron and a positron with $p_t{}^e > 3$ GeV/c each with the full size TRD ($|\eta| < 0.9$) from Υ decay is $\varepsilon_{geo} = \frac{N_{accepted}}{dN/dy \cdot \Delta y} = 0.65/\Delta y$ where Δy is the width in rapidity of the uniform input distribution (see also Chapter 12). For the reconstruction of resonances two electrons need to be found and identified in coincidence. The reconstruction efficiency for Υ therefore reads

$$\varepsilon_{det} = (\varepsilon_{act} \cdot \varepsilon_{tracking} \cdot \varepsilon_{PID})^2,$$

with ε_{act} - the effective active fraction of the solid angle, $\varepsilon_{tracking}$ - the efficiency to find a track that hits the active area and ε_{PID} the survival probability of electrons passing the PID cut (see section 6.4.5). The product $\varepsilon_{act} \cdot \varepsilon_{tracking}$ is shown in Fig. 6.35 for single tracks, ε_{PID} is targeted to be 90%, at which a pion efficiency of $\varepsilon_{PID}^{\pi} = 0.02$ (for dN/dy = 8000) can be reached in the offline analysis from the TRD data alone (see Section 11.5).

Table 6.8: Integral number of Υ under various conditions for an integrated luminosity of $\mathcal{L}_{int}^{PbPb} = 0.5/\text{nb}$. Cut A of Table 6.6 was employed (* using Cut B).

	N_{T8000}^{Υ}	N_{T2000}^{Υ}	N_{MB8000}^{Υ}	$N_{T8000}^{\Upsilon}(Muon)$
Maximum charged particle multiplicity dN/dy	8000	2000	8000	8000
produced number of Υ, N_{int}^{Υ}	285000			
decaying into acceptance	24750			15019
decaying into acceptance without TPC pile-up	12672			-
reconstructible with OFFLINE efficiency	4041	4437	4041	9349
reconstructed with standard options	1056	2016	154	8414
reconstructed with TRF off, 30 time bins	3053	4052	-	-
reconstructed with deconvolution of TRF	2269	(4180)* 3628	-	-

The single track efficiencies for the different scenarios as shown in Fig. 6.35 and described in Section 6.4.4 have been parameterized by straight lines. The resulting differential distribution of reconstructed Υ resonances as function of centrality for a multiplicity density of 8000 for central collisions is shown in Fig. 6.39. Note that the number of produced Υ grows like $N_{part}^{4/3} \propto (dN/dy)^{4/3}$. The lower histograms in Fig. 6.39 reflect the reconstruction efficiency and its drop with increasing multiplicity. The solid grey histogram is obtained by employing offline efficiencies at the trigger level (including the cut on single particle transverse momentum of $p_{\rm t} > 3$ GeV/c) and follows the production probabilities. For the more realistic online trigger scenarios a reduction, especially for central collisions, is observed. The corresponding integral numbers over the full centrality range are given in Table 6.8 in the column labeled N_{T8000}^{Υ}. Under those conditions about 2300 reconstructed Υ can be expected per year of ALICE running (employing the deconvolution scheme of the TRF). This number has to be compared to an untriggered scenario where the whole analysis is performed offline. When still aiming at the full centrality coverage as motivated by the physics case, the total available bandwidth to the dielectron physics (assumed to be 20 Hz of central events) would be filled with minimum bias events (74 Hz due to the smaller event size). The number of reconstructible Υ for this scenario has to be down-scaled by the ratio of DAQ rate to minimum bias interaction rate. The values are shown in Table 6.8 in the column labeled N_{MB8000}^{Υ}. About 150 Υ could be reconstructed per year of ALICE running. One could trade performance in centrality coverage for more statistics in central collisions under the constraint of the same DAQ bandwidth. An extreme option would be to give up the centrality coverage completely and focus on central events only. In the 10% most central events about 1600 Υ can be found with offline efficiencies under the conditions used for Table 6.8. According to Table 6.7 the central events occur with a frequency of 195 Hz. Given the bandwidth of the DAQ system of 20 Hz, 164 Υ can be reconstructed in such an event sample. These numbers clearly demonstrate that the TRD dielectron trigger is essential for this physics topic.

The numbers of reconstructed Υ can be substantially improved by exploiting the capabilities of the HLT system [15]. This system is designed to allow for an input bandwidth of 200 Hz. Assuming that half of that bandwidth is allocated for central event processing (for the other half see Section 6.5.3) 100 Hz of central events can be inspected. With offline detection performance 820 Υ can be gathered within one ALICE year. This number has to be compared to the number of Υ that is already in the triggered sample,

e.g. 680 for the scenario with tail cancellation. The running mode with a full scale HLT TRD analysis delivers very similar total Υ numbers as is given by the central subset that is included in the minimum bias TRD triggered event sample. While the numbers are comparable for central events, it is, however, not possible to cover the centrality dependence of Υ production with the HLT scheme due to the rate limitations of opening the TPC gating grid.

The inspection of central collisions by the HLT analysis can be used in addition to increase the number of reconstructed Υ: combining both modes of data taking a final number of 1160 Υ in central collisions can be achieved in one ALICE year.

As discussed in Chapter 12, the extrapolation of latest RHIC results predict a rapidity density of $dN/dy \approx 1600$. While the beam energy increase is too large to allow a reliable extrapolation, we will still consider 2 scenarios for the discussion of the trigger performance, as the implications for the performance of the TRD trigger system are significant. The numbers corresponding to a multiplicity in central events of dN/dy_{cen} = 2000 are listed in the column labeled N^{Υ}_{T2000} in Table 6.8. The number of reconstructed Υ is almost doubled for the scenario that employs the tail cancellation technique. Additionally about 15% can be gained by opening the selection cuts (Cut B from Table 6.6) leading to a total of 4200 triggered Υ.

Note that complementary efforts to detect Υ states in Pb-Pb collisions yield similar numbers. When using the same production cross section and scaling to the same integral luminosity of $\mathcal{L}^{PbPb}_{int} = 0.5/\text{nb}$ per year, the CMS experiment would reconstruct an integral number of 6400 Υ per year [12]. The performance of the ALICE muon arm is given in Table 6.8 and is expected to be still better.

The numbers presented in Table 6.8 correspond to the full size TRD detector without any holes. Should the active area have to be reduced to 50% of the nominal solid angle, a reduction of the measured signal to about a quarter of the numbers in Table 6.8 would be the consequence (see also Chapter 12). The physics program of exploring QGP properties with Υ states would be severely limited if the TRD could not have complete coverage.

6.5.3 Trigger rates

The true Υ trigger rate is very small (10^{-2} Hz). The rate of the TRD dielectron trigger is dominated by background and has contributions from 3 sources:

1. misidentified pion tracks
2. fake tracks from combination of clusters from different primary tracks
3. true electron tracks due to conversion of photons before the TRD

The single track background multiplicities are shown in the center and right panel of Fig. 6.35. The center panel represents the sum of all the background contributions. In the right panel the background from true electrons (component 3 from above) is separately plotted, since for this component no reduction can be achieved from the PID. For the components 1 and 2 the pion suppression factor ξ shown in Fig. 6.37 was used. The total background multiplicity M_{bck} can then be constructed from the total found tracks M_{found} and the found electron tracks $M_{found-e}$ by $M_{bck} = (M_{found} - M_{found-e})/\xi + M_{found-e}$. For the scenario employing the tail cancellation (see Fig. 6.35) this background multiplicity $m(M)$ was parameterized as function of centrality M by a square root behavior. The background consists of tracks of positive and negative charges with equal probability. Due to the statistical nature of the background tracks the distribution is assumed to be Poissonian and the differential background pair trigger rate R_{pair} is given by

$$\frac{dR_{pair}}{dM} = \frac{dR_{mb}}{dM} \cdot (1 - e^{-m(M)/2})^2,$$

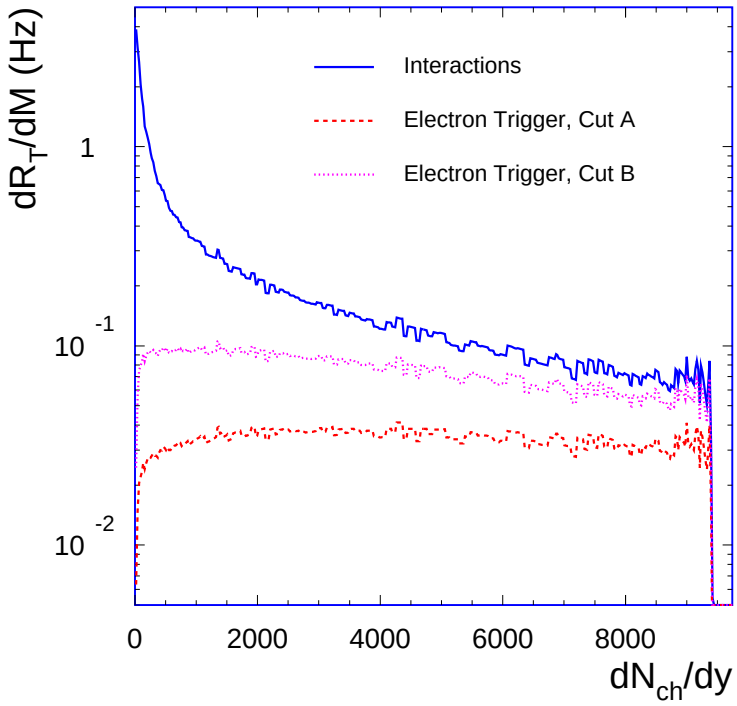

Figure 6.40: Differential background trigger rate R_T as function of centrality for Pb+Pb at 5.5A TeV with a multiplicity for central events of dN/dy = 8000. Different Cuts listed in Table 6.6 were used to select stiff electron candidates.

where R_{mb} is the minimum bias interaction rate. The differential trigger rate as function of centrality given in terms of dN_{ch}/dy is shown in Fig. 6.40 for the two sets of parameters listed in Table 6.6. The integral numbers are shown in Table 6.9 in the rows labeled 'e^+e^-'.

Table 6.9: Trigger rates for Pb+Pb at 5.5 A TeV.

$dN/dy_{central}$		2000	8000
Cut A	e^+e^-	128 Hz	351 Hz
	Quarkonia	43 Hz	117 Hz
Cut B	e^+e^-	340 Hz	732 Hz
	Quarkonia	113 Hz	244 Hz

A further reduction of the trigger rate is achieved by calculating the invariant mass of the electron pair in the GTU. As was found in the Technical Proposal (Fig. 38) [3] the background spectrum is essentially flat over the invariant mass range from 1 to 10 GeV/c^2. With the transverse momentum resolution as given in Fig. 6.32 an invariant mass resolution of 8% at the Υ mass is achieved by the TRD trigger. Selecting only 2 intervals of invariant mass around the J/ψ ($2.8 < m_{e^+e^-} < 4.0$ GeV/c^2) and the Υ family ($m_{e^+e^-} > 8$ GeV/c^2) reduces the total trigger rate by about a factor of 3, while the physics signal is reduced only by about 10% as can be seen from Fig. 12.7. The corresponding integrated rates are given in Table 6.9 in the rows labeled 'Quarkonia'.

The parameter values for 'Cut A' ('Cut B') were tuned at the expense of some efficiency loss to have an acceptable trigger rate even for the multiplicity scenarios $dN/dy_{cen} = 8000$ ($dN/dy_{cen} = 2000$) for central collisions. So, in each case the dielectron trigger rate would be about 110 Hz. Such a trigger rate into the High Level Trigger (HLT) seems to be appropriate. The current HLT design foresees a maximum

input bandwidth of about 200 Hz [15]. The 'Quarkonia' dielectron trigger would occupy one half of the total available bandwidth. Further optimization of the online background strategies is currently under investigation. The numbers given here should be taken as orientations of what can be achieved already with very simple algorithms.

An overall suppression of the event rate by more than a factor 10 is expected from an offline like analysis in the HLT compute farm. The task of the HLT is to further sort out the correct events by:

1. Applying the full pion suppression power with a maximum likelihood analysis of the pulse height distributions (see Section 11.5). Since the online pion rejection was assumed to be only 1:20 independent of centrality at least a reduction by a factor of 2 can be expected from this step.

2. Tracking of all candidates through the TPC and ITS to remove fakes and conversions. Approximately 60% of the electron background tracks originate from particles with transverse momenta below 3 GeV/c . The curvature and the closest distance to the vertex are much better determined by the global tracking. The latter quantity can also be used to reduce the other 40% of the background, i.e. stiff tracks not originating from the primary vertex. In total, a minimal reduction by a factor of 2 is anticipated. A detailed analysis of the global tracking performance of the central ALICE arm is in progress.

Such a reduction of the single candidate multiplicities by a factor of 4 results in a reduction of fake pairs by a factor of 16 resulting in an output trigger rate of the HLT system is in the order of 7 Hz for the 'Quarkonia' trigger scenarios '$dN/dy_{cen} = 8000$, Cut A' and '$dN/dy_{cen} = 2000$, Cut B'. For the lower multiplicity scenario the 'e^+e^-' trigger without invariant mass cut represents an alternative option compatible with the allocated DAQ bandwidth.

Bandwidth requirements to the data acquisition system can be further reduced by considering partial readout of regions of interest. The stiff tracks will cross only 3-4 out of 2×18 TPC sectors. Therefore a reduction by about 10 is possible. Such a readout scenario will not allow for a detailed cross correlation of the dielectron signal with global TPC quantities. This could be an option for an intermediate time period when possibly the HLT is still under study and development.

Summarizing, the selection criteria of the combined TRD / HLT dielectron trigger can be adjusted such that the output rate to DAQ is in the order of 10 Hz when triggering on the quarkonia states and about a factor 2 higher without the invariant mass selection. The bandwidth can stay well below the anticipated limit of the equivalent of 20 Hz of central events of $dN/dy_{cen} = 8000$.