PRC 00/05

H1 Fast Track Trigger
Specification of the L1 System

H1 Collaboration

October 11, 2000

Abstract

The H1 collaboration has been asked to present to the October PRC meeting a "close to full" simulation of the first level part of the Fast Track Trigger algorithm, including the current understanding of the prospects for providing information to the level 1 H1 central trigger. This document contains the requested information, including a near to final specification of the full level 1 system. The L1 trigger option has been studied in detail. As a result it has now been decided to include the L1 trigger option in the design.

1 Introduction

The H1 Fast Track Trigger (FTT) project is described in detail in [1]. The FTT is expected to provide information to the level 2 (25 μs) and level 3 ($\sim 100\ \mu$s) triggers. During the design phase of the FTT it was recognized that the device could also be used to improve the first level track triggering in H1[1] with only a marginal increase in cost [2, 3]. This 'FTT L1 Trigger Option' would improve the signal to background ratio for most physics processes. A notable and simple example is the elastic production of light vector mesons (e.g. ρ, ϕ) for which the vector meson decay products result in two charged tracks, often with low p_t. A fully integrated L1 trigger algorithm design and simulation of its implementation in hardware are now in place.

For this the following tasks have to be implemented within the level 1 latency of 2.3 μs:

- the digitisation of drift chamber signals
- hit detection
- determination of pulse arrival times and z-coordinates
- search for patterns of hits on groups of three wires consistent with track segments originating from the vertex region
- linking of track segments from 4 different groups and generation of a trigger decision

[1]The existing 'DCrϕ' L1 track trigger has two loose thresholds of $p_t \sim 400$ MeV and $p_t \sim 800$ MeV. The FTT will have significantly improved p_t resolution, allowing variable thresholds for $p_t > 100$ MeV. Unlike the DCrϕ trigger, the FTT is also expected to provide precise track multiplicity information.

The track segment finding algorithm [3] has not changed conceptually since the last PRC presentation. However, its implementation in hardware has been investigated using simulation tools and the resulting estimates of speed and required resources have formed the basis for the hardware specifications [4].

This note begins with a short overview of the main L1 tasks and their planned implementation in hardware. Each functional step is then described in detail. Resource allocation and timing estimates from the simulations are also given. A summary of manpower needs and costing are given at the end.

2 FTT L1 Overview

A flow chart of the main operation steps in the level 1 algorithm can be found in fig. 1. The left hand branch of this figure shows the steps taken before a L1 trigger decision (L1KEEP) is received from the central trigger (pre-L1KEEP). The processing in this branch is repeated for each bunch crossing with a specific hypothesis for the bunch crossing of origin (t_0). The final L1 trigger signals have to be sent to the central trigger within approximately $2.1\ \mu$s of the event t_0. The right hand branch shows the related tasks to be performed after a L1KEEP signal (post-L1KEEP) is received from the central trigger (generated $2.3\ \mu$s after the t_0 such that the bunch crossing of origin is specified). No stringent time limits are imposed on this post-L1KEEP processing. Wherever possible, the same hardware is used in the two branches. Detailed descriptions of each block shown in fig. 1 are given in sections 3 and 5.

Fig. 2 shows the overall system architecture of the FTT, covering all three trigger levels. The crate layout for the L1 system and an overview of the Front End Module (FEM) layout is given in fig. 3. Fig. 4 shows schematically the FEM and the main tasks to be performed in it. A short summary of the algorithm is given in this section.

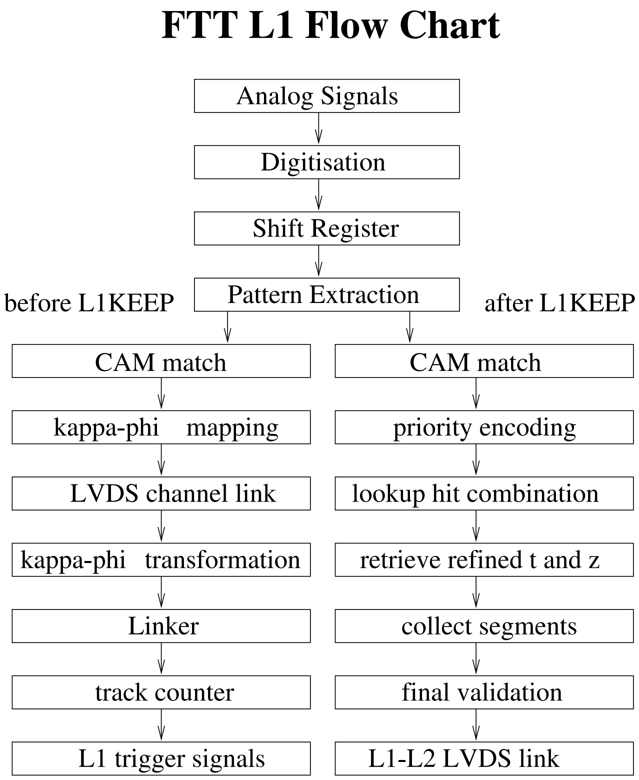

Figure 1: Flow chart of the functional steps of the L1 track finding algorithm.

Analogue Cards (top of fig. 4) tap selected drift chamber wires and feed the analogue signals into the FEM. The data are digitised using FADCs. The information from groups of three wires are then processed by ‘front’ FPGAs (Altera APEX 20K600) in which the hit and segment finding takes place. The $Q\&t$ analysis performs a hit recognition, determines the charge collected at each wire end and extracts the z-coordinate using the charge division technique. Hits are sampled at 80 MHz, though for the initial stages of the algorithm and for L1 triggering, groups of four adjacent time slices are ORed and the hits are passed to shift registers synchronised at 20 MHz. Track segment identification takes place by matching the contents of the three shift registers (one per wire) to predefined acceptable patterns stored in Content Addressable Memories (CAMs). In the pre-L1KEEP phase, this matching takes place for each bunch crossing. In the post-L1KEEP phase the shift registers are frozen. After correcting for the fixed delay between the t_0 and the arrival of the L1KEEP signal, the corresponding regions of the shift registers are sent to the same CAMs.

The algorithm now divides into the two branches corresponding to the pre- and post-L1KEEP cases as indicated in fig. 1. Before L1KEEP, the data follow the arrows in fig. 4 into the I/O controller. Track segments from several FEMs are then sent via an LVDS channel link to Merger Cards and finally to the L1 Linker card. Here track segments from trigger layers at different radii are matched to identify tracks using a sliding window technique in $\kappa(\propto 1/p_{\mathrm{t}})$-$\phi$ space. The L1 Linker hardware is identical to the L2 Linker card to be built by SCS [5] since the L1 and L2 linking algorithms are very similar. Finally, fully linked tracks are counted and L1 trigger signals (Trigger Elements) to be sent to the central trigger are formed.

After a L1KEEP signal is received from the central trigger, the shift registers are halted and the CAM match for track segment finding is repeated. The finer granularity 80 MHz sampling information is retrieved for each hit and the data follow the arrows in fig. 4 into the L2 validation block of a second smaller FPGA (Altera APEX 20K400). Here, it is checked that the track segments remain valid in light of the 80 MHz information and refinements are made to the κ and ϕ values using this information. These steps are realized using look-up tables implemented in external SRAMs. Finally the refined and validated track segments are transmitted via the LVDS channel link to the L2 system.

3 Description of FTT L1 Trigger algorithm

In the following, a functional description is given of all steps required for the L1 trigger decision (top and left branch of fig. 1). The steps in the post-L1KEEP phase (right branch) are discussed in section 5. The mapping of the algorithm onto the front and back FPGAs and the number of bits of information per segment passed between the different stages are further illustrated in fig. 5.

3.1 Tapping of drift chamber signals at existing electronics

Analogue cards collect the drift chamber signals from the input to the existing DCrϕ track trigger system on the front of the CJC FADC crates. A block diagram of the design can be found in fig. 6. The Analogue Cards drive (driver AD3138) the pre-amplified analogue signals of a single trigger group (3 wires read out at both ends) up to $5\ \mathrm{m}$ to the FEM cards. This “plug through” operation produces minimal disruption to the existing H1 configuration. The functioning of the DCrϕ trigger will not be jeopardized, so that the FTT L1 and DCrϕ triggers can operate simultaneously during the commissioning phase of the FTT.

3.2 Digitisation

Signals from 5 trigger groups adjacent in the ϕ coordinate are fed to one FEM, resulting in a total of 30 input analogue signals per board. These signals are digitised using dual 8-bit linear FADCs (AD9288) operated at

80 MHz clock frequency. Signals from both ends of each wire are digitised in a single circuit, such that the effects of electronic noise on the subsequent charge division process are minimised. The digitised data are passed to an FPGA (APEX20K600) farm, where the subsequent processing to produce track segments takes place.

3.3 $Q\&t$ analysis

The first step in the FPGAs is the $Q\&t$ analysis. Hits are detected using a DOS (difference of sample) technique as described in [3]. A hit is defined by the clock cycle of the first maximum in the DOS. Two hits can be distinguished if they are separated by more than 50 ns.

In contrast to the description given in [3] it is now planned to extract the z coordinate separately for each individual hit, rather than obtaining only one z coordinate per track segment. The new scheme improves the robustness of the z coordinate extraction. In addition passing the full z information for each wire to the L2 system allows a more flexible r-z fit with the possibility of outlier rejection.

A schematic drawing of the revised $Q\&t$ algorithm can be found in fig 7. The charge at each end of a wire is integrated over several bunch crossings using a running summer [3]. The z coordinates, determined by charge division, are associated to hits at a later stage of the algorithm (see section 3.4). The z coordinates are stored in 6 bits of information per wire, corresponding to a digitisation in bins of 3.4 cm. With the present gain, this bin size is smaller than the intrinsic z-resolution even for highly ionising particles.

3.4 Shift Registers

The timing output from the $Q\&t$ algorithm is synchronised at 80 MHz clock frequency. For the first stage of the algorithm, groups of four successive 80 MHz digitisations are ORed in order to create shift registers synchronised at 20 MHz. The shift registers at a given clock cycle can be thought of as a representation of the positions of the hits in the chamber for a particular t_0 hypothesis. The shift registers are analysed once per bunch crossing, such that every possible t_0 hypothesis is investigated. In total five shift registers are filled per ϕ cell. Three shift registers contain information from the three wires of the the main cell and two further shift registers contain information from a selected wire in the cell immediately to the left and right. Track segments crossing cell boundaries can thus also be recognised.

Since the $Q\&t$ algorithm provides hits separated by at least 50 ns, only one hit is possible per 20 MHz time slice. The location of the hit in the four corresponding 80 MHz time slices can thus be encoded in two bits. This encoded information, which is later used for refinement purposes after L1KEEP, is stored in a 20 MHz shift register of two bit depth. By setting a fixed delay, the z coordinate information is associated with the corresponding hit and is stored in a further 20 MHz shift register of 6 bit depth.

3.5 Pattern Extraction

A search for valid track segments in the r-ϕ plane takes place next. Valid track segments are defined as groups of three hits, one in each of the three shift registers, consistent with a track originating from the vertex region. The search is realised by dividing the shift registers into groups of 'patterns'. A pattern comprises short sequences of bins grouped together from each of the three shift registers, representing broad roads through which genuine tracks may have passed. These patterns are presented to CAMs, where valid segments are identified.

3.6 CAM match

The search for valid track segments within a pattern is realised using Content Addressable Memories (CAMs). The CAMs store the predefined valid hit combinations across the three registers. Invalid combinations do not have entries in the CAMs. Where a hit combination from the registers matches one of those stored in the CAM, the corresponding output line goes high. There is a one to one relation between output line and stored hit combination. The CAMs are operated in the unencoded multiple match mode, such that if more than one valid hit combination exists within a pattern, all corresponding output lines go high simultaneously.

A sufficient number of CAMs is implemented to store up to 2048 hit combinations per cell. That limit allows all possible segments to be found for any track with $p_{\rm t} > 100\ \mathrm{MeV}$, including cases where tracks cross cell boundaries within a group of three wires. Because the CAM match is performed every bunch crossing, all possible hit combinations for all possible t_0 hypotheses are considered.

3.7 κ-ϕ mapping

The next step is to assign κ and ϕ values to each track segment found in the CAMs. For the level 1 FTT trigger, the full $\kappa \times \phi$ space is represented in a 8×60 grid for each track sign. Each half CJC1 cell or full CJC2 cell corresponds to a single ϕ bin.[2] With the help of the vertex constraint, each valid segment can be mapped uniquely to a bin in the 8×2 histogram in $\kappa \times \phi_i$, where i refers to the trigger layer. The mapping differs between trigger layers. The assignment of each track segment to a bin position in the $\kappa \times \phi_i$ histogram is hardwired by ORing together all CAM outputs corresponding to the same κ and ϕ_i bin. The final output to the I/O control in fig. 4 is the unencoded $\kappa \times \phi_i$ histogram for a particular CJC cell.

3.8 LVDS channel link and Merger Cards

The κ-ϕ_i histogram blocks from all CJC cells are sent via Merger Cards to the L1 Linker card using the LVDS channel link. On the merger cards the 8×2 (8×1) $\kappa \times \phi_i$ histogram blocks from CJC1 (CJC2) are merged together within a single trigger layer to produce a global histogram of size 8×60 in $\kappa \times \phi_i$.

3.9 κ-ϕ transformation

The $8 \times 60_i$ track segment representation for each trigger layer has to be transformed to a global κ - ϕ_{glob} representation for the linker card. This is achieved by routing in a FPGA the appropriate input pins and internal input lines to the linking algorithm. After transformation the theoretical κ - ϕ_{glob} bin center positions of the different layers do not perfectly match. These unavoidable shifts do not harm the linking algorithm (section 3.10) because the sliding window technique used takes that into account automatically.

3.10 Linker

Linking track segments with compatible κ and ϕ in different trigger layers is realised by looking for coincidences in the same or nearby bins in the κ - ϕ_{glob} histogram, using a sliding window technique (currently 3×3 in size). At least 2 matching track segments out of the maximum possible 4 are required. Although individual track segments are usually valid for many bunch crossings, the segments can often only be successfully linked when the t_0 hypothesis is correct.

[2] The ϕ coordinates are defined at the track apex, rather than the vertex, in order to ensure this direct mapping between CJC cells and ϕ bins.

3.11 Counting of track multiplicities and generation of Trigger Elements

The result of the linking step is a 8×60 κ - ϕ_{glob} histogram with entries wherever tracks have successfully been linked. The Trigger Elements to be sent to the central trigger are derived from the contents of this histogram. The Trigger Elements can be based, for example, on a single high p_t track or on track multiplicities with a variety of programmable p_t thresholds. Topological criteria (e.g. back to back coincidences) can also be applied. The exact definition of the Trigger Elements requires input from the Physics Working Groups and can be changed at short notice. A possible set is shown in table 1:

L1TE#	condition	L1TE#	condition
0	Any track (event t_0)	10-12	summed track charge $\sum Q$
1,2,3	0-7 tracks $p_t > 100$ MeV	13	back to back topology
4, 5	0-3 tracks $p_t > 200$ MeV	14	1-jet topology (tracks close in ϕ)
6, 7	0-3 tracks $p_t > 400$ MeV	15	2-jet topology (tracks close in ϕ)
8, 9	0-3 tracks $p_t > 1.0$ GeV		

Table 1: Possible definitions for the 16 Trigger Elements to be sent to the central trigger by the FTT L1 system.

4 Status of the L1 Trigger Implementation

The status of implementation of all L1 Trigger related tasks is summarised in table 2. All hardware components are in or near the design stage with the exception of the trigger interface card, which merely converts different signal levels. The backplane specifications are almost finalized. Many aspects of the algorithms running on FPGAs have been simulated with specialised software [6] and are now being coded in VHDL. A full 'top down' simulation has been written for the FEM tasks from the shift register creation to the $\kappa - \phi$ mapping. The linking and track counting algorithms to be implemented in the L1 linker card have been simulated and are ready for implementation.

5 Description of the Post-L1KEEP Step

After a positive L1 decision by the central trigger, the L1KEEP signal is distributed to all branches. This signal arrives at the FTT a fixed time ~ 2.4 μs after the bunch crossing of origin, such that the exact event t_0 is implied. After receiving the L1KEEP, all FTT pipelines and shift registers are halted. The track segment finding as a pre-processing step for the L2 linking and fitting operation then commences. The pattern extraction is repeated using a fixed region of the shift registers shifted by about 1.2 μs (24 bins at 20 MHz) from the region considered at the pre-L1KEEP stage. Patterns are formed in an identical manner to that described in section 3.5.

5.1 CAM match

The search for valid segments in the CAMs is the same as that used for L1-triggering (section 3.6) as is the number of valid hit combinations (up to 2048).

Module	**Function**	**Comment**	**Status**	**Responsible**
Analogue Card	Provide analogue DC signals	existing path should not be affected	design phase specification complete	Manchester
FEM	hardware		specification ready fine tuning needed analogue simulation	RAL
	$Q\&t$ analysis	FPGA	C simulation on existing data in progress. Existing VHDL to be reviewed	RAL
	Shift Register	FPGA/RAM	coded in VHDL simulated	B'ham/ETH
	Pattern Extraction	FPGA	coded in VHDL simulated	B'ham/ETH
	CAM match	FPGA/CAM	coded in VHDL simulated	B'ham/ETH
	κ-ϕ mapping	FPGA with	dummy routing coded in VHDL simulated	B'ham/ETH
Merger Card	hardware		design phase specification complete	SCS/ETH
	data collection	FPGA	FORTRAN simulation VHDL coding to be done	SCS/ETH
L1 Linker Card	hardware		design phase specification complete	SCS/ETH
	L1 linking	FPGA	coded in VHDL simulated	B'ham/ETH
	Generation of Trigger Elements	FPGA	VHDL coding to be done	B'ham/ETH
Service Module	STC interface	HERA clock or PLL	specification being finalised	Manchester
Backplane	custom VME/ hard metric	common L1/L2 backplane	specification phase contact with Wiener	Manchester
Trigger Interface	signal interface		to be designed	Manchester or B'ham

Table 2: Overview of L1 trigger components, their functionality, the design status and the responsibility.

5.2 Priority Encoding

Up to this point, the processing of different regions of the shift registers has been performed in parallel. For the remaining tasks, it is necessary to switch to serial processing. 'Priority encoding' circuits are used to serialise the valid segments output from the CAMs. The addresses of valid hit combinations emerge from the priority encoder in sequence, ordered by address. This does not necessarily imply any ordering of segments by importance, though that functionality may be used for example to give priority to high p_t track segments in high multiplicity events. The 11 bit addresses, selected from the list of ≤ 2048 possible valid segments, are passed to a First In, First Out (FIFO) buffer.

5.3 Look-Up of hit combination

During CAM operation, the register positions of the three hits which generate the valid match are lost. This information is needed (section 5.4) in order to obtain the refined t and the z information stored in parallel 20 MHz shift registers. The register positions of the three hits generating the CAM match is restored using a RAM look-up table. Since there is a one-to-one relationship between the CAM output line and the stored information in the RAM, this architecture is called a 'tag field'.

5.4 Retrieve refined t and z information

The next step is to re-associate the finer granularity 80 MHz sampling and the z coordinates of each hit with the track segment found in the CAMs at 20 MHz sampling frequency. The shift register positions of the three hits extracted in the tag field (section 5.3) are used to look up the encoded 80 MHz timing information and z coordinates from the shift registers described in section 3.4. This information is later used to test the validity of each segment in fine detail and to refine the κ and ϕ values (see section 5.6).

5.5 Collection of track segments and refinement information

The 11 bit valid mask number and the associated 80 MHz refinement information and z coordinates are passed from all five front FPGAs to a single smaller FPGA at the back of the FEM (see fig 5). This FPGA serves as I/O controller and also performs the final validation of the track segments.

5.6 Final validation of refined track segments

Before the track segments are sent out to the FTT L2 system, their validity is tested in light of the refined 80 MHz hit information. In this step about 50% of the track segments are rejected because the more precisely known hit combination cannot be extrapolated to the primary vertex position. For each segment, 17 bits corresponding to the 11 bit 20 MHz mask number and the 3×2 bit 80 MHz refinement information are fed to an external SRAM where a 'validation' bit is added, indicating whether the track segment is kept or rejected. If it is kept, a second look-up table in the SRAM is used to obtain the remaining track parameters which are needed by the level 2 system for linking and fitting.

The list of valid refined masks and corresponding (κ, ϕ) coordinates are periodically determined using recent calibration constants and are loaded into the external SRAMs at times when data is not being taken (between HERA fills or H1 runs). For this purpose, it is planned to install an online calibration system, which continuously monitors essential parameters such as the beam position and the drift velocity in the chambers.

Finally, the full validated and refined track segments are built. The information to be passed to the L2 FTT comprises the curvature κ and slope ϕ of the segment, the precise location (x-y coordinate) of the hit on the central wire and the z coordinates of all three hits. The final track segments are buffered in this form in FIFOs.

5.7 L1-L2 LVDS channel link

The validated and refined track segments are buffered and serially transmitted via the Merger cards to the level 2 system using the same channel link as used for L1-triggering.

6 Status of the Implementation of post L1KEEP tasks

The status of implementation of all tasks to be performed after the L1KEEP is listed in table 3. Only those tasks which are not common with the pre-L1KEEP processing are listed. The largest resource usage comes from the priority encoding. This has been simulated in detail. The retrieval of the 80 MHz information and z coordinates have been coded in parts. The buffering and SRAM access tasks to be performed on the the back FPGA have not yet been coded, though these are standard operations which will not present any problems.

Module	Function	Comment	Status	Responsible
FEM				RAL
	Priority encoding	front FPGA	coded in VHDL simulated	B'ham/ETH
	Buffering of segments	front FPGA/FIFO	partly coded in VHDL partly simulated	B'ham/ETH
	Look-Up hit combination	front FPGA/RAM	VHDL coding to be done	B'ham/ETH
	retrieve z refined t	front FPGA/RAM	partly coded in VHDL partly simulated	B'ham/ETH
	Collect Segments	back FPGA/FIFO	VHDL coding to be done	B'ham/ETH
	Final validation	back FPGA/SRAM	VHDL coding to be done	B'ham/ETH

Table 3: Overview of the FEM tasks after receiving the L1 trigger signal (L1KEEP). Also shown is the design status and the responsibility.

7 Timing and Necessary Resources

7.1 Timing before Level 1 Keep

Estimates of the times taken to perform all tasks up to the sending of Trigger Elements to the central trigger are given in table 4. The numbers in the table should be compared to the latest point at which the Trigger Elements can be sent to the central trigger ($\sim 2.1\ \mu$s). The longest delay arises from the maximum drift time to the chamber sense wires (about $1.1\ \mu$s). In addition to this, the delays due to cable lengths, digitisation, data transmission between cards and Trigger Element transmission to the central trigger are also unavoidable. The quoted delays related to the pipelining of the algorithm were obtained by simulation of the individual tasks. The uncertainty in the simulations is of the order of a few clock cycles (10-12 ns) and is thus of similar size to variations in the maximum drift time due to changing atmospheric pressure.

After accounting for all other steps, less than 20% of the total L1 latency remains for the track finding algorithm and Trigger Element definition logic. Estimates of the time taken for these last steps indicate that the algorithms just fit into the available time. The VHDL implementation which has been used to simulate the timing delays has not yet been optimised and a few tens of ns may also be saved by optimising the L1-L2 data transmission.

Since the timing is tight, a fall-back solution has also been investigated, whereby hits from the longest drift times ($> 1.0\ \mu$s) are ignored in the processing to produce L1 Trigger Elements. Such long drift times occur only in the outermost trigger layer of CJC 1 and represent only about 1% of all hits (see fig. 8). Cutting out these long drift times would thus save 100 ns without a significant loss in track finding efficiency. With the availability of this additional option, the FTT L1 trigger will be able to provide useful information to the level 1 trigger within the available time.

Task	**Latency (ns)**	**Cumulative time (ns)**	**Reference**
Ionisation + drift to sense wires	1100	1100	calculated / measured
Analogue Cable Delays	180	1280	calculated / measured
T-Analysis + FADC delay	108	1388	VHDL simulated / specification
write shift register + synchronisation	48	1436	VHDL simulated
CAM match	24	1460	VHDL simulated
κ-ϕ mapping*	0	1460	VHDL simulated
transmission and κ-ϕ transformation	≤ 410	1870	calculated and simulated [7]
Linking + trigger decision	≤ 182	≤ 2052	VHDL simulated
trigger transmission	50	≤ 2102	calculated

Table 4: Delays of the different tasks before a L1-Trigger decision is taken. Shown are the individual delays, the cumulative time assuming a maximum drift time of 1100 ns and a reference for how the delay has been evaluated. *Note that the hardwired routing for the κ-ϕ mapping adds only a small signal transmission delay of about 1 ns and does not add an additional clock cycle.

7.2 Timing after Level 1 Keep

The time constraints for the post-L1KEEP processing are nothing like as critical as those for the L1 Trigger Element production. Optimisation of FTT L2 algorithms and the increased speed of the LVDS channel link [3] have significantly reduced the L2 timing estimates. Recent results on the implementation of tasks in the L2 FTT have shown that the segment linking and track parameter optimisation can be finished after about $14\ \mu$s, well inside the L2 latency of about 20μs. Some of the remaining $6\ \mu$s will be used to implement simple particle resonance searches such as elastic vector mesons at L2. However, several hundreds of ns will certainly be available for preprocessing at L1.

Table 5 shows a list of the timing delays in the L2 preprocessing to be performed in the post-L1KEEP phase. The first five tasks are implemented in the front FPGAs. The numbers quoted for these tasks are taken from the present simulation results with pipelined algorithms and a clock of 80 MHz driving the CAM validation. Tasks to be performed in the validation and refinement steps on the back FPGA cannot be simulated in full by the Quartus software, since they involve external SRAM circuits. The Micron datasheet on ZBT SRAMs [8] has been used as the basis for the time estimates for these tasks, assuming a pipelined SRAM access at 100 MHz

[3] It has been decided for L1 Trigger purposes to run the link at 100 MHz instead of 50 MHz as originally planned

(133 MHz is the maximum speed). The 100 MHz frequency matches the clock speed used for the L1-L2 link.

Task	Latency (ns)	Time first segment (ns)	Time last segment (ns)
Receive L1KEEP	80	80	80
Read Shift Register + CAM match	24	104	104
Priority Encoding	24	$\leq$200	424
Buffer segments	24	$\leq$224	448
Look-Up hit combination	24	$\leq$248	472
retrieve refined t and z	24	$\leq$272	660
Collect Segments	24	$\leq$296	684
FPGA-FPGA transmission	60	$\leq$356	744
Validation in SRAM	40	$\leq$396	1186
Write to FIFO	10	$\leq$406	1196

Table 5: Calculated timing delays for the tasks that are performed in the post-L1KEEP phase. 'Time first segment' defines the delay after which the first track segment can be sent out to the L2 system. 'Time last segment' defines the time when the last segment is processed. A maximum number of 80 track segments can be sent out to the level 2 system from a single FEM within 1196 ns. All time evaluations have been made by using VHDL simulations with the exception of the final validation in the SRAM and the FPGA-FPGA transmission.

As can be seen from the table, the delay after L1KEEP is at most 406 ns before the first track segment can be sent to the L2 system. The 80th segment would be sent after about 1.2μs[4]. That result fits well with the L2 linker algorithm, which expects a maximum of 128 track segments from all FEM cards of the same trigger layer (6 FEM cards for CJC1 and 12 for CJC2) within 1.3 μs of the start of transmission. We conclude that there will be no problems arising from time taken by the L1 algorithms after L1KEEP.

For completeness the current understanding of the L2 timing is given in table 6.

task	chip	frequency	steps	max. time
L1 track segment finding	L1 system			$\sim$ 0.5μs
receive data / fill CAMs	FPGA	100 MHz	$\leq$ 130	1.3 μs
linking of $\leq 4 \times 128$ segments	FPGA	100 MHz	$\leq$ 528	(+2.6 interleaved) 2.6 μs
fitting of $\leq 2 \times 24$ tracks	DSP	166 MHz	1060	6.4 μs
trigger decision	DSP, FPGA	166 MHz	$\sim$ 500	3.0-9.0* μs
total				13.8-19.8 μs

Table 6: Timing estimates for each individual L2 task. Note that linking and fitting are interleaved. The actual trigger decision time depends on the number of invariant mass combination to be calculated at level 2. *Basic trigger decision parameters like transverse momentum thresholds, topological criteria or the invariant mass of two prong final states can be calculated within 3 μs. The L2 trigger capabilities can also be extended to calculate invariant masses in multi-particle final states. In that case the full remaining 9 μs would be used up.

[4]Note that the final number of 1.2μs cannot directly be compared with table 3 in [3], since the order of steps, specifically the collection of track segments from 5 adjacent cells and the final validation, has been reversed in order to save SRAMs.

7.3 Required FPGA Resources

In contrast to the timing considerations, the biggest requirements on FPGA resources arise from the post-L1KEEP tasks. The biggest Altera chips available on the market are the APEX20K400, 600, 1000, and 1500 which integrate about 400k, 600k, 1M and 1.5M gates respectively. These chips are pin compatible, such that final decisions on which FPGAs to use can potentially be left until the production phase of the FEM. For the FEM design, we presently assume that we will use the APEX20K600, with the possibility to use the APEX20K400 to save money if the algorithm can be fitted into it. The APEX20K600 has 24320 logic cells (registers) and 154 Embedded System Blocks (ESBs) which are used to implement RAMs (2 kbit per ESB) or CAMs [3]. It is planned to clock most parts of the algorithm at 80 MHz, which must be compared with the maximum clock speed computed by the simulation for each step in the algorithm. It has been found that the maximum clock speed depends on the synthesizer software. For all studies in this note, results obtained using the Quartus software have been taken. Some of the results have been cross-checked using other software packages. No optimisation procedures have been tested or applied so far. Potential for optimisation is given by so called ”back-annotating” the design and by changing routings at the register level. In addition, regular improvements in the development software (usually every 3-6 months) lead to better optimised design implementations. For the current study, the Quartus version 2000.05 from Altera [6] has been used. An improved version is expected to be released in October 2000.

7.3.1 Resources of the front FPGA

The Resource usage in the front FPGA on the FEM board is listed in table 7. The number of logic cells and ESBs used by each step in the algorithm is given as obtained by VHDL simulations. Those steps labelled ”proved by VHDL simulation” are in agreement with the expectations based on theoretical considerations.

About 65% of the logic cells of the 20K600 FPGA are used up by the L1 algorithm. Detailed studies have shown that a degradation of the maximum speed occurs if more than 70% of the FPGA resources are used. Even before any optimisation procedure, the algorithm is found to fit comfortably onto the 20K600 FPGA. Additional logic may be implemented for control purposes etc. at the level of a few hundred logic cells. This is expected to fit without a significant degradation of the performance. Timing analyses of the current design showed that the 80 MHz clock requirements are fulfilled with a safety margin of about 30%.

Task	Number logic cells	Number ESBs	reference
$Q\&t$ analysis	$< 3000^*$	0	proved by VHDL simulation
Shift Register	1920	5	proved by VHDL simulation
Pattern Extraction	200	0	proved by VHDL simulation
CAM match	4800	64	proved by VHDL simulation
κ-ϕ translation	688	0	proved by VHDL simulation
Priority Encoding	4000	0	proved by VHDL simulation
Buffer segments	400	4	proved by VHDL simulation
Look-Up hit combination	0	16	proved by VHDL simulation
Retrieve refined t and z	~ 400	0	estimated
Collect segments	100	2	proved by VHDL simulation
Control and IF	< 500	0	estimated
total	< 16008	91	

Table 7: Resource usage of all tasks performed in the front FPGA on the FEM board. *The existing implementation of the $Q\&t$ analysis is going to be tuned and reduced in size.

7.3.2 Resources of the back FPGA

The main tasks of the back FPGA are the collection of data from the five front FPGAs and the validation of track segments using the look-up tables in the SRAM. Because the back FPGA also serves as I/O controller, the FPGA size is mainly determined by the number of required I/O pins. It was decided to use an APEX20K400 with 488 user pins.

7.3.3 Resources of the L1-Linker Card FPGA

The L1-Linking algorithm is a modified version of the L2-linking algorithm and has been adapted from L2 without major changes for a κ-ϕ histogram of size $8\times$ 60 bins. Although the implemented algorithm uses for first level trigger purposes a rather big sliding window with 3×3 bins to search for track segment matches, [5] the full algorithm fits into an APEX20K400 FPGA with about 16000 logic cells. The implemented linking algorithm uses 63% of the logic cells and no ESBs. The current implementation of the algorithm can be clocked slightly above the planned operation frequency of 100 MHz. Optimisations are foreseen to increase the maximum clock speed and improve the safety margin.

Now that the feasibility of the "L1 Trigger option" has been demonstrated, further studies are required in order to find the optimal linking algorithm. Refinement of the VHDL code, taking into account any new findings, will then take place. Because the current implementation of the algorithm fills only 63% of the 4th biggest FPGA on the market it is clear that FPGA resources are not a critical issue here. The generation of Trigger Elements from the linked tracks has not yet been considered in detail. That task will add at most a few hundred logic cells without affecting the resource demands significantly.

8 Manpower and Financing

Table 8 shows a list of all active persons in the design and construction of the FTT. In two cases, *no name* (N.N.) is allocated to uncovered tasks, namely the implementation of the readout and of the calibration and monitoring. The group is in the process of searching for two students to cover these tasks. With these exceptions, the manpower situation at all system levels is sufficient.

The total cost of all levels of the FTT has been estimated at 1730 kDM, of which 1480 kDM are currently approved. 200 kDM have been requested from the German BMBF for the year 2001 and a decision is expected next year. 50 kDM have been requested from a German-Polish fund for supporting infrastructure (crates, PCs, etc.). A decision is expected this year. Currently the FTT is within budget although being confronted with high exchange rates for purchasing FPGA circuits in US$. ETH Zürich made a contract in August with the SCS company for the construction of the L2 boards. The financing of the design and prototyping is assured. The series production of boards, planned for summer 2001, cannot commence until the remaining 250 kDM has been approved.

9 Summary

A basic design for the level 1 FTT is now in place. Most aspects of the algorithms have been fully simulated, such that FPGA resource and timing requirements have been accurately estimated. The small number of missing pieces of information are expected to be available in the near future. All hardware components and their

[5] A 2×2 sliding window in place of 3×3 would reduce the amount of logic by about 50%. This option is currently being investigated.

Institute	Person	Engineer	Diploma	PHD	Postdoc	task
Birmingham	P.Newman				X	L1
	Y.Fleming			X		L1
	R.Staley	X				L1
Manchester	S.Kolya				X	L1
	D.Mercer	X				L1
RAL	D.Sankey				X	L1
	A.Baird	X				L1
	B.Claxton	X				L1
Zürich	A.Schöning				X	FTT
	D.Meer			X		L2
	N.N (calibration)			X		FTT
SCS	company	3				L2
Dortmund	C.Wissing			X		L2
	J.Naumann			X		L3
	M.Kolander	X				L3
	O.Behrendt		X			L3
DESY	I.Cheviakov				X	L3
	J.Spalek				X	L2/L3
	N.N (readout)			X		FTT

Table 8: List of all people actively involved in the FTT project. Also shown is the distribution of tasks. N.N. stands for two Phd students the project is currently looking for. SCS stands for the company *Supercomputing Systems* in Zürich which is building the boards for the L2 system.

interfaces have been specified. The option of sending trigger information to the level 1 trigger has proved to be viable, such that the present architecture incorporates this option. The design and layout of electronic boards, backplane and other aspects of the hardware is underway. In parallel with this, work will continue on the optimisation of algorithms.

References

[1] 'A Fast Track Trigger with High Resolution for H1', and addendum, PRC 99/06, PRC 99/07.

[2] 'A Fast Track Trigger with High Resolution for H1', Proposal to the PPESP, number 750.

[3] 'Level 1 Fast Track Trigger Feasibility Study', H1 FTT group, April 2000.

[4] FTT Specification Workshop, Dortmund, 26/27th September 2000.

[5] 'H1 Second Level Fast Track Trigger Feasibility Study', Supercomputing Systems (2000).

[6] Quartus version 2000.05, http://www.altera.com/html/tools/quartus.html.

[7] FTT L1 internal documentation, PRC companion, October 2000.

[8] 'Designing with ZBT SRAMs', Micron TN-55-01, Rev. 9/99.

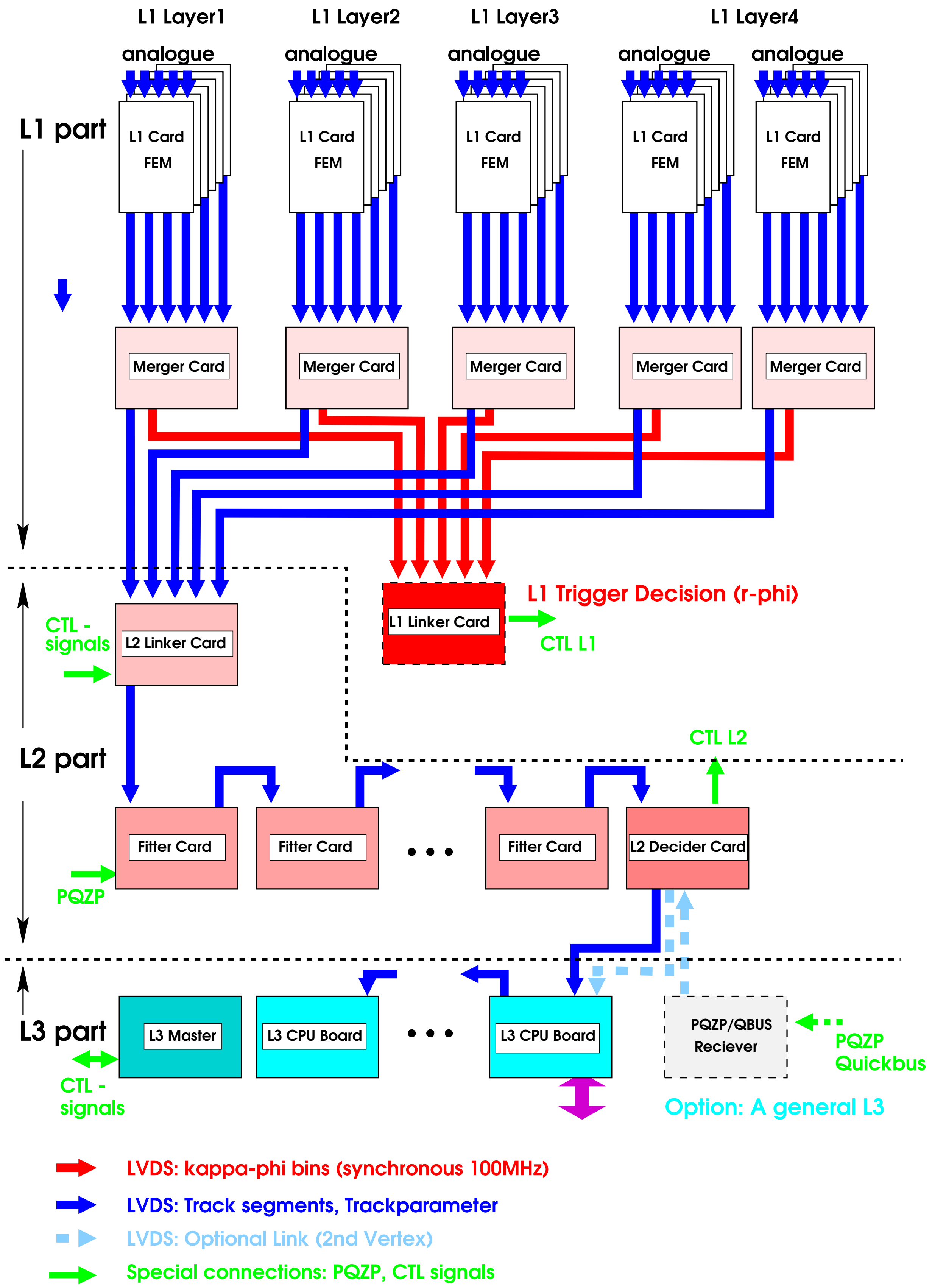

Figure 2: Schematic overview of the FTT hardware. Analogue Cards are not shown.

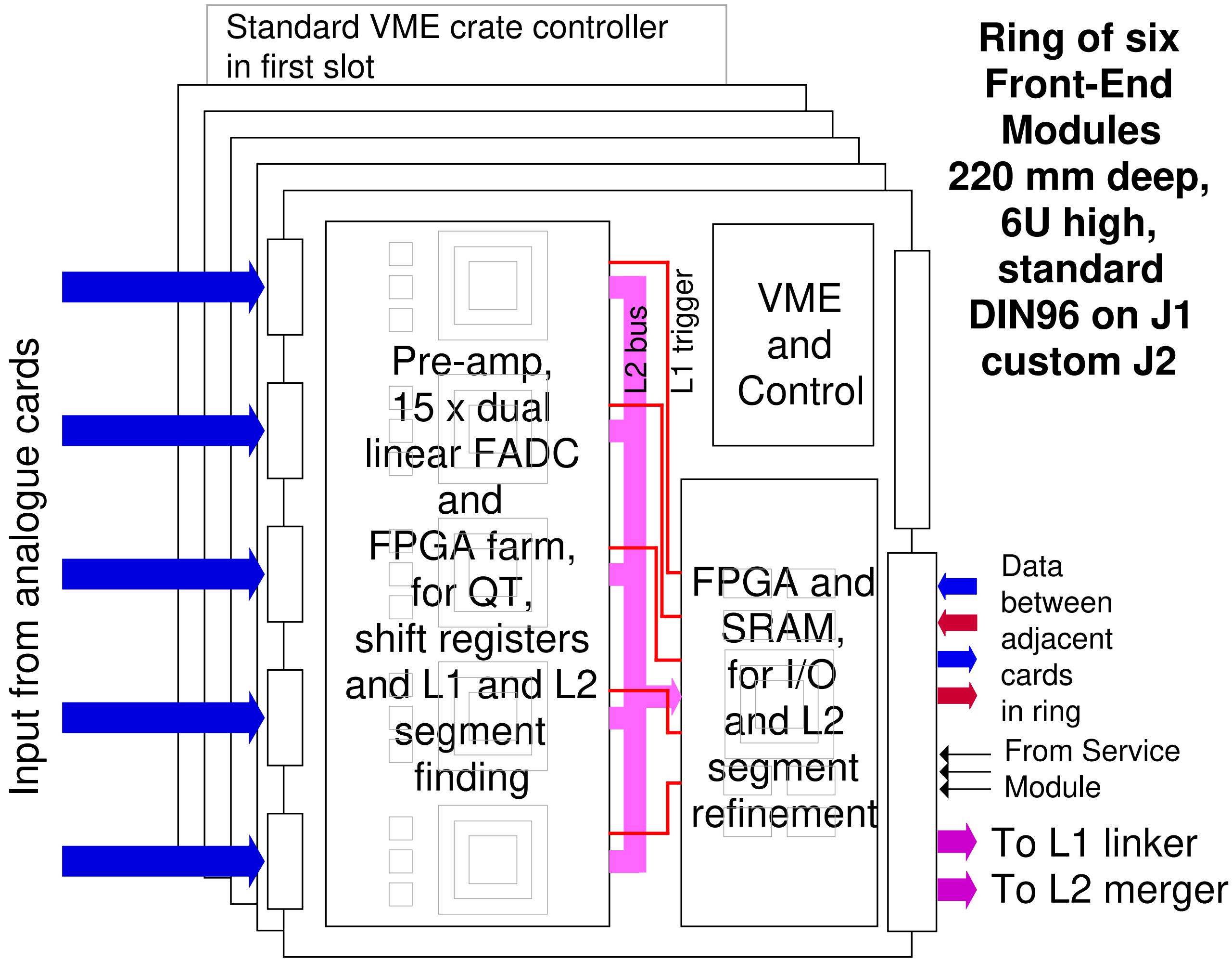

Figure 3: Schematic overview of the L1 crate filled with FEMs

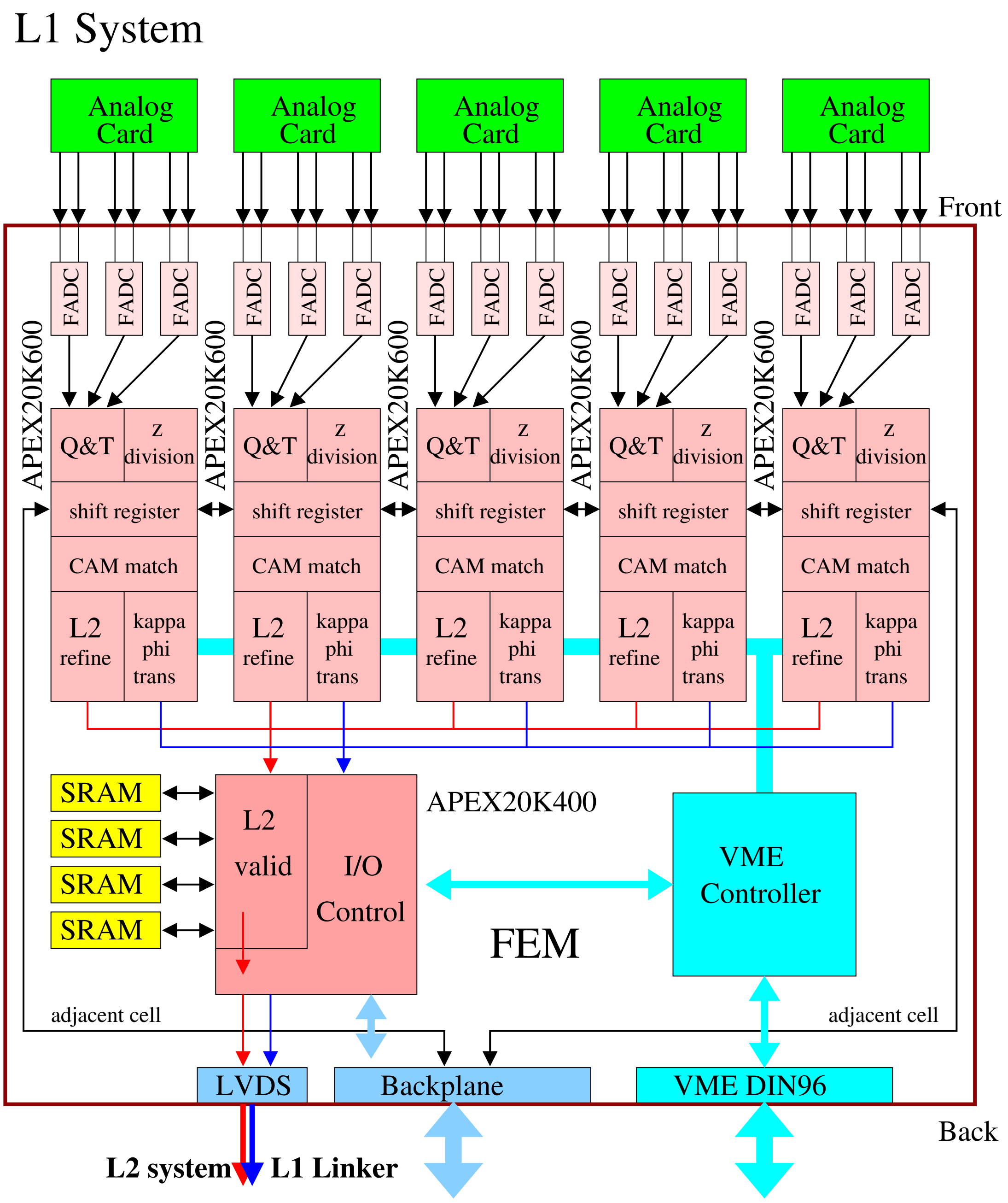

Figure 4: Block diagram showing the tasks performed in the Front End Module

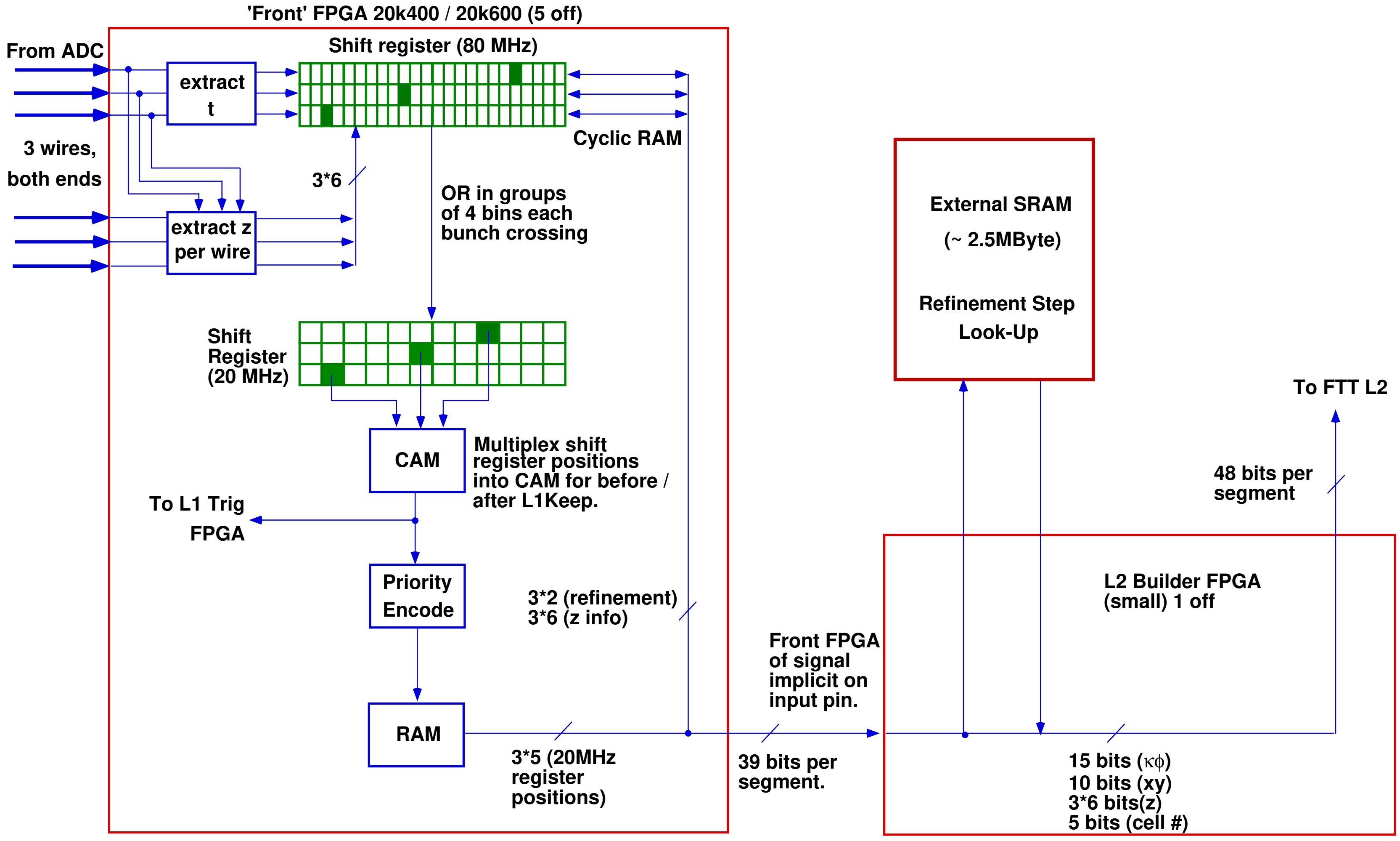

Figure 5: Schematic overview on the FTT L1 algorithm.

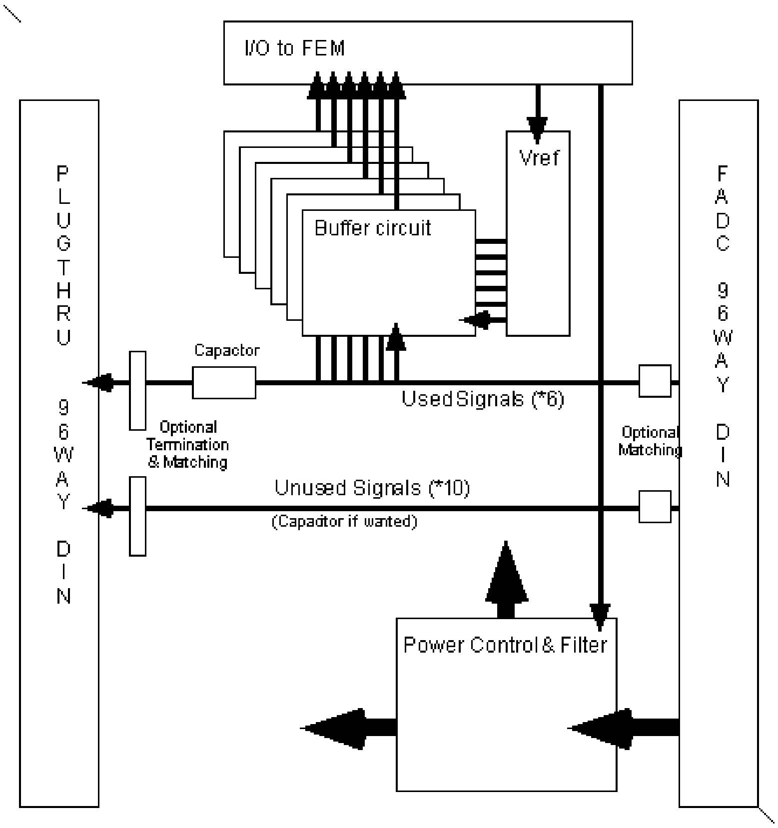

Figure 6: Schematic design of the Analogue Card

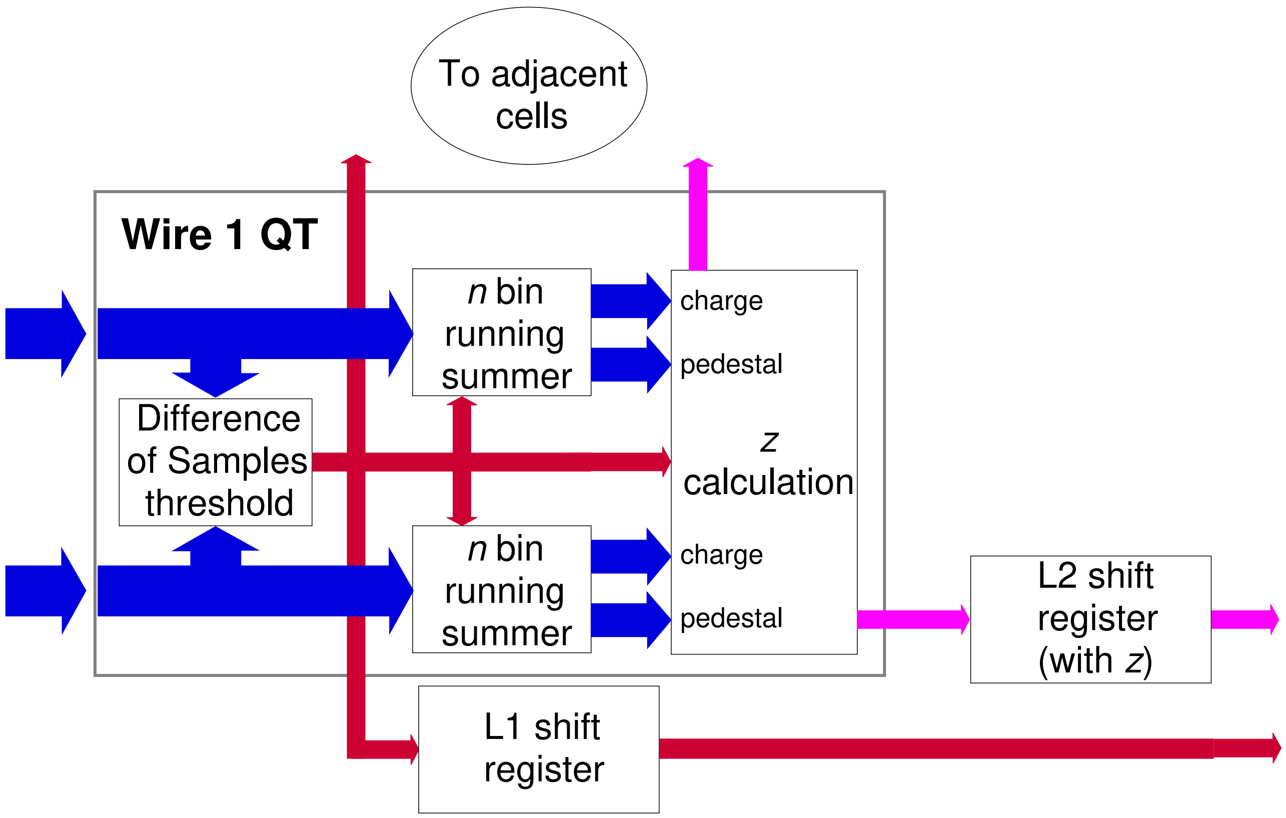

Figure 7: Schematic overview of the $Q\&t$ algorithm as implemented in the front FPGA.

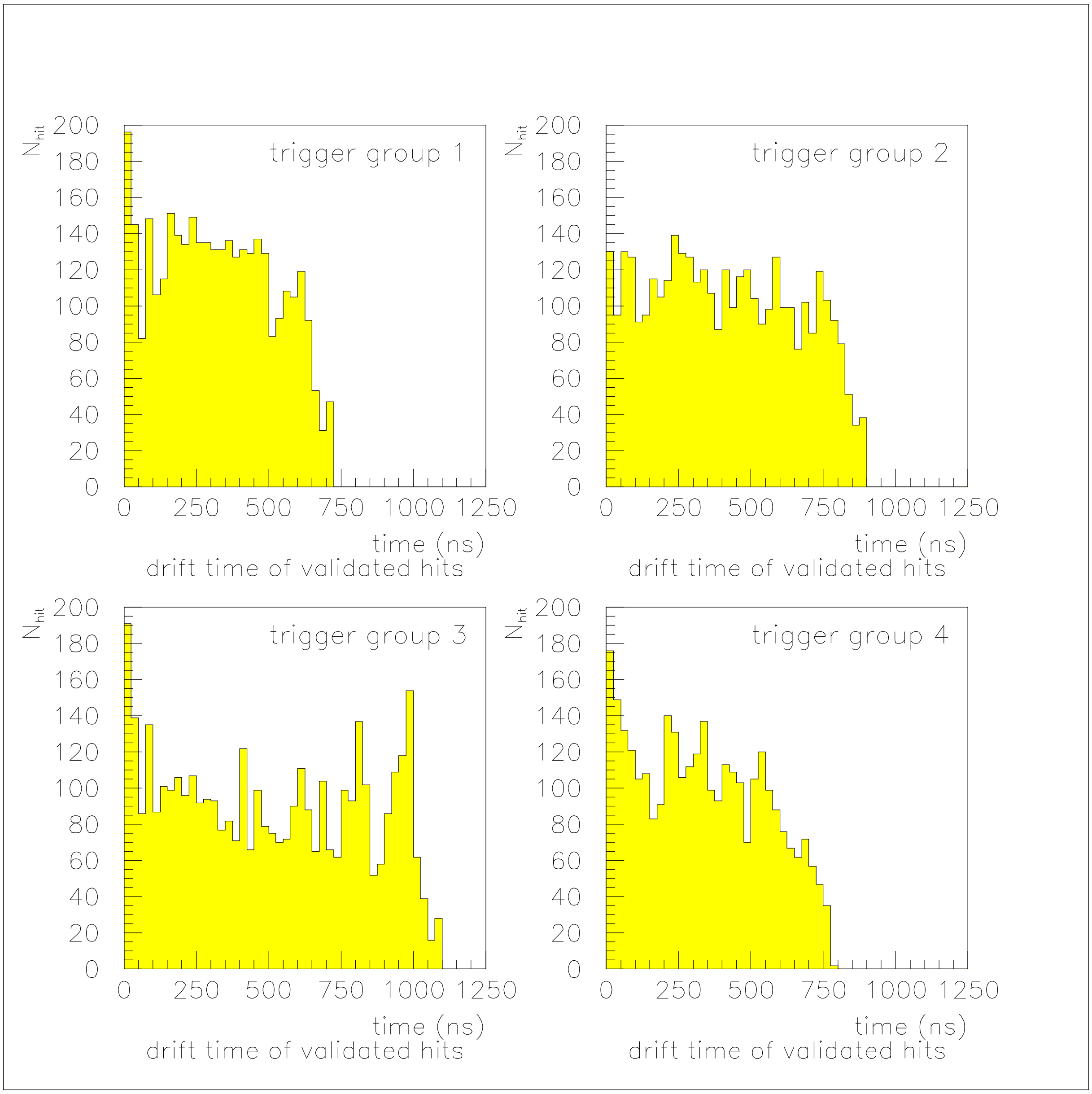

Figure 8: Drift time distribution of hits used by the FTT and validated by the track segment procedure. Data from 1997 have been taken. The drift time is shown for the outermost wires of the four trigger layers.