

A Massively Parallel Track-Finding System for the Level 2 Trigger in the CLAS Detector at CEBAF

David C. Doughty Jr., Penny Collins, and Stephan Lemon

CHRISTOPHER NEWPORT UNIVERSITY
50 Shoe Lane, Newport News, VA, 23606, USA

Peter Bonneau

CONTINUOUS ELECTRON BEAM ACCELERATOR FACILITY
12000 Jefferson Ave., Newport News, VA, 23606, USA

Abstract

The track segment finding subsystem of the LEVEL 2 trigger in the CLAS detector has been designed and prototyped. Track segments will be found in the 35,076 wires of the drift chambers using a massively parallel array of 768 Xilinx XC-4005 FPGA's. These FPGA's are located on daughter cards attached to the front-end boards distributed around the detector. Each chip is responsible for finding tracks passing through a 4 x 6 slice of an axial superlayer, and reports two 'segment found' bits, one for each pair of cells. The algorithm used finds segments even when one or two layers or cells along the track is missing (this number is programmable), while being highly resistant to false segments arising from noise hits. Adjacent chips share data to find tracks crossing cell and board boundaries. For maximum speed, fully combinatorial logic is used inside each chip, with the result that all segments in the detector are found within 150 ns. Segment collection boards gather track segments from each axial superlayer and pass them via a high speed link to the segment linking subsystem in an additional 400 ns for typical events. The Xilinx chips are ram-based and therefore reprogrammable, allowing for future upgrades and algorithm enhancements.

I. INTRODUCTION

The distinguishing feature of the CEBAF facility is its ability to provide a continuous beam of 4 GeV electrons to three experimental areas. The beam is actually composed of microbunches of electrons separated in time by two-thirds of a nanosecond. Each experimental endstation receives every third bunch, leading to a 500 MHz bunch rate of electrons on target. Two of the endstations have focusing spectrometers (high resolution but low acceptance), while the third contains a large acceptance device called CLAS (CEBAF Large Acceptance Spectrometer).

The CLAS detector is designed for kinematic analysis of several particles in the final state of nuclear interactions. It uses a toroidal magnetic field generated by six coils for momentum analysis; for this reason the detector package has been partitioned into six wedges or sectors. Each sector fits between two adjacent coils and consists of four types of detectors: six superlayers of hexagonal cell drift chambers for

charged particle tracking, Cerenkov detectors for electron-pion separation, scintillation counters for particle identification by time-of-flight (TOF), and an electromagnetic calorimeter (ECAL) for energy measurements of electrons, neutral pions, and photons. Three of the drift chamber superlayers have wires perpendicular to the midplane of the sector; these are called the axial superlayers. Track positions along these wires are measured by the three stereo superlayers, whose wires are skewed by 6 degrees relative to the axial wire direction. These six superlayers are grouped into three pairs, with one axial and one stereo superlayer in each pair. Details of the detector design are given in reference [1]; a side view of two opposite sectors showing the placement of the detectors is shown in Fig. 1.

Charged particles emerging from the target traverse the innermost superlayer pair before entering the region of high magnetic field. The toroidal field then bends them toward or away from the beam, depending on the sign of each particle. The middle superlayer pair measures the track trajectory during this bending phase, while the outer superlayer pair measures the outgoing trajectory after the particle has left the region of high magnetic field. Combining the measurements from all six superlayers allows us to determine the initial track trajectory (given mostly by the inner superlayer pair) and the momentum of the particle (determined by the curvature in the magnetic field which is measured by the middle and outer superlayer pairs).

The CLAS detector is designed to run at luminosities exceeding $10^{34} \text{ cm}^{-2}\text{s}^{-1}$ producing a hadronic interaction rate of several Megahertz. The data acquisition system is being designed to handle an event rate of up to 10 kHz. To acquire desired events with high efficiency while minimizing the deadtime, a two-level hierarchical trigger has been designed. The LEVEL 1 trigger is deadtimeless, processing all prompt signals through a three-stage pipelined memory lookup within 90 ns. Details of the design of the LEVEL 1 trigger are given in reference [2]. The resulting signal provides a common start signal to the photomultiplier tube ADCs and TDCs; a delayed version (to allow for the drift time) is used as the common stop for the drift chamber TDCs. After the LEVEL 1 trigger accepts an event the detector is dead for 2 μs . During this time the LEVEL 2 trigger uses hit information from the drift chambers to find tracks and match them with the trigger

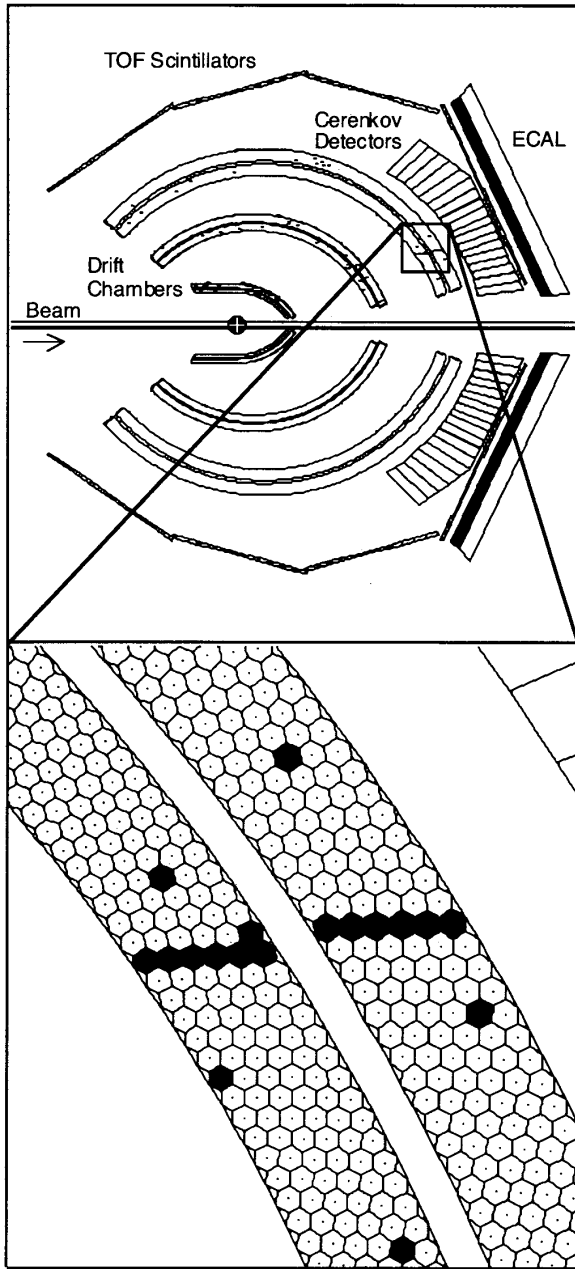


Figure 1. A side view of two opposite sectors of the CLAS detector showing the arrangement of the six drift chamber superlayers, and the Cerenkov, TOF, and ECAL detectors. Dots in the drift chambers in the upper figure are cells which fired either because of noise, or because a particle passed through the cell as shown in the expanded lower figure.

requirements. Once an event is accepted at LEVEL 2, conversion of the front end data is initiated and the detector

will not go live until all ADCs and TDCs have digitized and locally buffered their data, a process which typically takes 20 μ s. If the event is rejected by the LEVEL 2 trigger, a fast clear and reset of the entire detector occurs. The LEVEL 2 trigger is the subject of this paper.

II. LEVEL 2 TRIGGER OVERVIEW

There are many demanding requirements for the LEVEL 2 trigger. Because the detector is not live during the time it takes LEVEL 2 to reach a decision, the trigger must be as fast as possible to minimize deadtime. A time limit of 2 μ s has been set for LEVEL 2, yielding a deadtime contribution of 2% at a LEVEL 1 trigger rate of 10 kHz. Cosmic rays can be a major source of LEVEL 1 background triggers, due to hits in the TOF or ECAL scintillators. These events must be rejected by the LEVEL 2 trigger, using the direction of the tracks in the drift chambers. The ability to select the momentum and polar angle of the tracks used to define the trigger is a desirable feature, as is the possibility of specifying the electron signature in the Cerenkov or ECAL detectors. Because the CLAS torus will sometimes be run with reduced or reversed magnetic fields, the LEVEL 2 trigger must be capable of being reprogrammed to handle these different fields. Extended targets up to 20 cm long will be used with CLAS, so the trigger must find and classify tracks originating from the entire extended target region. It must be as close to 100% efficient as possible to virtually eliminate event losses, and minimize the number of false triggers caused by noise or out-of-time event fragments. Finally, it would be desirable to be able to modify the trigger in the future as actual experience is gained as to the nature of the noise in the detector.

To simultaneously meet these demanding requirements, the LEVEL 2 trigger uses a massively parallel array of 768 field programmable gate arrays (FPGA's) to simultaneously find track segments in all cells of the axial superlayers (the stereo superlayers are not used in the LEVEL 2 trigger). Track segments from each axial superlayer within a sector are collected and passed to the linking subsystem. The six linker modules (one per sector) use an array of associative memories to match stored patterns of valid track 'roads' against the segments found in the event. The address of the road is then used as the input to a lookup table which gives the momentum and angle of the track, and predicts its location in the other detectors. A list of all found tracks with correlation information from the other detectors is sent from each sector to the LEVEL 2 Event Processor. The Event Processor then compares the found tracks with the specified triggers, and produces the final trigger signals which are sent to the trigger supervisor.

The granularity of the track segments is a crucial parameter of this system. Finer segmentation leads to slightly better resolution in momentum and angle when the segments are linked, but increases the data handling problems and the number of roads required for linking. A CEBAF developed event generator and detector simulation package called SDA [3] has been used to simulate the LEVEL 2 trigger. (This package

accounts for the energy loss and multiple scattering of a particle as it traverses the detector, but not interactions or decays.) A segment width of two cells yields a manageable number of roads (less than 10,000), while giving good momentum resolution. Figure 2 shows the momentum resolution which can be obtained in the forward direction (scattering angles of 30-50 degrees) for tracks with momenta in the range of 0.3-4.0 GeV.

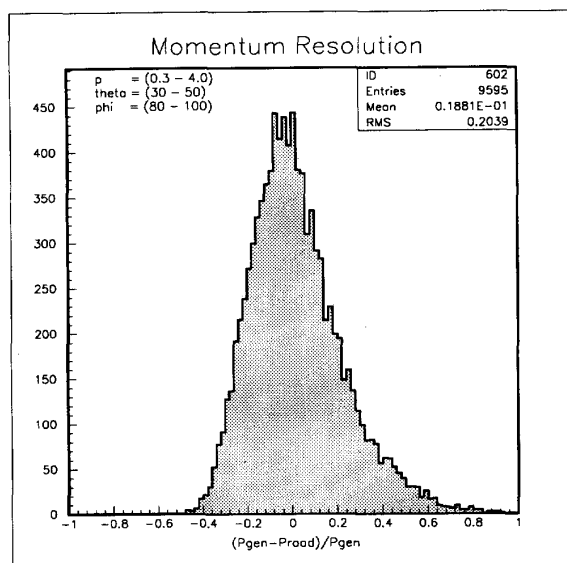


Figure 2. The momentum resolution obtained by linking track segments two cells wide in each of the axial superlayers. Events are in the forward direction (scattering angle of 30-50 degrees) with momenta between 0.3 and 4.0 GeV.

III. LEVEL 2 SEGMENT FINDING

A. Requirements

The algorithm used for finding the track segments largely determines the performance of the LEVEL 2 trigger. If track segments are missed, the corresponding tracks will not be found. If false, noise induced, segments are found, the trigger takes longer to process the event (increasing the deadtime), and the linker may find extraneous tracks. For these reasons the segment finding subsystem must be nearly 100% efficient at finding track segments, allowing for some drift chamber inefficiency, while being highly resistant to noise hits producing false segments. These two requirements are somewhat in conflict and much effort was expended toward finding a solution which optimized both. For the linker to find low-momentum tracks, segments from tracks at angles up to 60 degrees with respect to the superlayer must be found. To fit within the time requirements mentioned above it must be extremely fast. The segment finding subsystem is allocated a

time budget of 600 ns; the other 1.4 μ s is reserved for the linking subsystem and the LEVEL 2 Event Processor. This 600 ns includes not only finding the segments, but also transmitting the found segment lists to the linker modules. Finally, it is desirable for it to be reprogrammable to allow for future upgrades or algorithm enhancements.

B. The Algorithm

The axial superlayers contain six layers of sense wires in a repeating hexagonal pattern, with each sense wire being surrounded by six field wires. There are 128 sense wires along each layer of the innermost axial superlayer, and 192 in the other two. Track segments are classified based on the cell traversed by a track as it crosses layer number four within the superlayer. To find tracks at angles up to 60 degrees, hits in a cluster of 31 cells (including the target cell in layer four) from all six layers must be examined. The segment-finding algorithm compares hits in these cells with several pre-defined track templates, and counts the number of layers with hits which match each template. The result is a number from zero to six which we call the layer count; this forms the basis of the segment finding algorithm.

For each target cell in layer 4, nine templates are used for finding track segments. These templates are specified in terms of cell pairs at the boundaries between layers 1 and 2, 3 and 4, and 5 and 6, within the superlayer. A cell pair is defined as two adjacent cells, in two different layers, crossing one of these boundaries. A template consists of one or more cell pairs at each of the same three layer boundaries. This group of cell-pairs at a layer boundary is called a cell pair cluster. Figure 3a shows the 31 cells which must be examined in order to find track segments through a given cell in layer four, at angles up to 60 degrees. It also highlights the three important layer boundaries, and illustrates the definitions of cell pairs and cell pair clusters. Figures 3b-3f show the four templates for right going tracks (there are four reversed templates for left going tracks) and the template for straight tracks.

This grouping of cell pairs into clusters at each boundary substantially reduces the number of templates necessary. This was important in reducing the amount of logic necessary to implement this algorithm in hardware. This does cause the system to be slightly more vulnerable to finding false tracks in a high noise-rate environment, but the effect is not significant in normal running.

The layer count for each target cell in layer four is obtained as follows. Each cell pair counts the number of hits in the two cells comprising that pair, and obtains either zero, one, or two. The cell pair within the cell pair cluster at each boundary with the largest number of hits is added to the corresponding cell pairs within the cell pair clusters at the other two boundaries, yielding a total layer count of from zero to six. Note that although layer four is used to classify the location of the track segment, it need not be present to find a track segment, and in terms of counting the number of layers is no different than any other cell in the template. This process occurs in parallel for all templates in the given cell, and if any

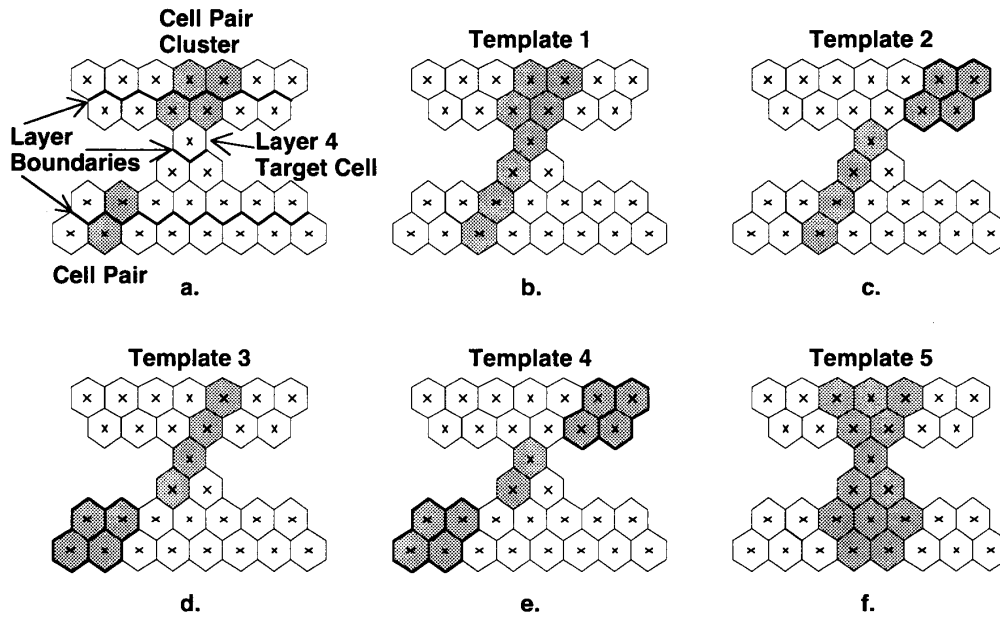


Figure 3. Figure 3a shows the layer boundaries and a cell pair and cell pair cluster, while Figures 3b-3f show the templates used for finding tracks in the superlayers. There are mirror images of templates 1-4 for left going tracks. In templates 2-4 cell-pairs within the wide angle clusters are outlined.

of the templates have a layer count of four or greater, this cell is declared to have found a segment with a layer count of four, five, or six, depending on the value of the highest layer count among its templates.

A single track passing through a superlayer will sometimes cause two adjacent cells in layer 4 to find segments with a layer count of at least four. To eliminate this problem of 'false doubling' a segment suppression scheme was implemented. A cell with a higher layer count suppresses an adjacent cell with a smaller count. In case of a tie, the cell on the left is suppressed.

The minimum number of layers required to find a segment is programmable from four to six, to allow for a range of inefficiencies in the drift chambers. Cells with layer counts which meet or exceed the programmed value have their track segment bits turned on. The logical or of the track segment bits from each pair of cells defines the two-cell-wide track segments.

This algorithm was developed and tested using simulated events in the CLAS detector. The simulation package SDA supplied the detector data. Initially, 10,000 clean tracks, with no noise and no missing drift chamber cells, and of various momenta and angle, were run through the algorithm. No track segments were missed in any of the superlayers.

To study this algorithm's robustness in cases of chamber inefficiency two data sets of 5000 tracks were generated. One of these contained positive tracks, the other negative, and each track had a 5% per layer probability of missing a hit. Because

the algorithm requires at least four layers or more to find a track it will miss those tracks in this sample with three or fewer hits. A simple calculation shows that it should miss 11.15 events out of each 5000. Running the simulation on these data sets yielded between 10 and 15 missed events in each superlayer, in good agreement with what was expected.

To study the vulnerability of this algorithm to finding noise-induced segments, events were generated with one track and with additional noise at the expected rate. The required layer count was set to four, to make the algorithm as susceptible as possible to finding extra segments. Because the noise rate is highest in the innermost superlayer, it has the highest probability of finding false segments. Figure 4 shows that under these conditions, less than 14% of the events have an extra segment found in this superlayer.

B. The Hardware

Signals from the sense wires are amplified by preamplifiers mounted on the chamber; groups of 96 channels (a 16 x 6 slice of each superlayer) are sent to each CEBAF designed front end board (FEB). These boards have three functions. They discriminate the input signals and produce an output which goes to a pipelined TDC. They also integrate the input; upon receipt of a delayed LEVEL 1 trigger signal they begin discharging at a constant rate, and produce another pulse when discharged. This effectively uses the pipelined TDC as a Wilkinson rundown ADC converter. Finally, two wires are

multiplexed into one TDC channel by using different pulse widths on the two channels, XORing the two outputs, and having the TDC record both the rising and falling edges of each pulse. More details on the operation of the FEBs is given in reference [4].

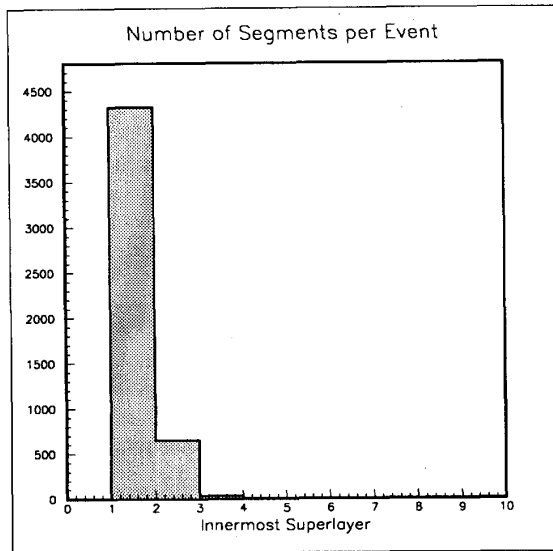


Figure 4. The distribution of the number of segments found in the innermost superlayer under typical noise conditions, requiring only four layers to define a track segment. A single segment is expected from the real track.

A crate of FEBs is controlled by a CPU in slot 0, which has an ethernet interface to the data acquisition slow controls system. The CPU communicates over a VME backplane to control the FEB boards. There are twelve FEBs, covering one superlayer, in the crate; the segment collector module sits in the middle of them. The trigger interface module brings in timing and control signals from the trigger system; these are distributed to the FEBs and the segment collector by a custom backplane. Figure 5 shows the layout of an FEB crate.

The FEBs have a flip-flop for each drift chamber cell which is set when that cell is hit. These 'hit bits' are used to find track segments according to the algorithm discussed above. A daughter module which plugs into the FEB looks for the eight track segments which could be present in the cells covered by this board. To do this it uses the 96 hit bits from cells on this board, as well as 14 hit bits from the FEB on its left and 11 hit bits from the one on its right. To implement the suppression part of the algorithm each daughter module communicates the layer count results (four, five, or six) from the two cells at the ends of its coverage of the superlayer, to the two adjacent modules. The backplane is designed to route both the hit bits and suppression logic signals between adjacent slots.

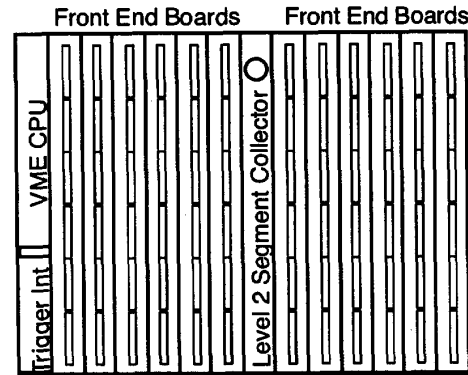


Figure 5. The layout of a crate of drift chamber front-end boards.

To implement the algorithm on the daughter card the Xilinx family of field programmable gate arrays (FPGA's) was chosen. These chips use static ram (SRAM) based cells to perform complex logic, requiring thousands of gates, in one chip. The latest generation of these chips has good density (to over 10,000 usable gates) and ample routing resources [5]. Because the logic is based on SRAM technology these chips must be configured at power-up, and there are several ways to do this. They may load data from an external EPROM (serial or parallel) or may have data pushed into them by an external controller. In our configuration we plan to use one serial EPROM per daughter card to configure all of the chips on that card in a master-slave configuration. This reconfigurability of these chips meets our requirement of allowing for future upgrades to the algorithm.

The logic for finding two track segments (covering four cells in layer 4) requires approximately 3100 gate equivalents, and fits into one XC-4005 chip, so that four chips on the daughter module cover the 16 x 6 slice of the superlayer handled by the FEB. The worst case pin-to-pin propagation delay predicted by the design tools is just over 90 ns. After allowing propagation time for adjacent module communication the segments should be found in well under the 150 ns goal.

A prototype of the daughter card was built using surface mount components. An automated test system was developed for this board using a Hewlett-Packard workstation and logic analyzer. Stimulus based on the templates was downloaded to the logic analyzer's pattern generator and applied to the chip inputs, while the outputs were observed with the state analyzer. The workstation uploaded the data and compared it with the predicted segments which should have been found. All expected track segments have been found, and the programmable layer count and suppression logic work. The measured worst-case propagation delay of 51 ns is shown in Fig. 6; this is considerably less than the 90 ns predicted. These chips appear to be considerably faster than predicted by the timing simulator. It is clear that this design will in fact

find all track segments in the CLAS detector in well under 150 ns.

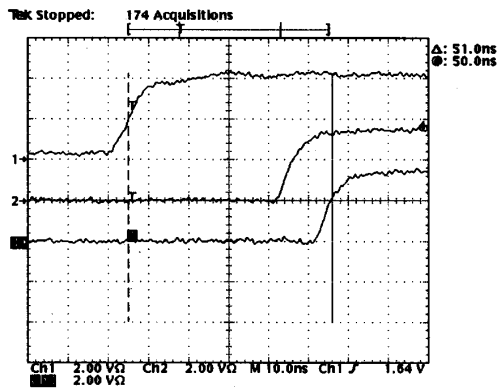


Figure 6. Measured worst case pin-to-pin propagation delay for the XC-4005 chip programmed with the segment finding algorithm. The top trace is the input, the middle trace is the suppression logic output, and the bottom trace is the found segment output.

IV. SEGMENT COLLECTION AND LINKING

A. Segment Collection

The segment finding daughter module on each FEB drives 8 output lines (corresponding to the 8 segments possible on this 16 x 6 slice of the superlayer) onto the backplane. The backplane routes all 96 of these lines (from the 12 FEBs which comprise the superlayer) to the segment collector module. Two state machines on the segment collector each encode all 'on' segment bits from the 48 bits in one half of the superlayer. By using full lookahead each is able to encode the segment addresses in one clock cycle of 40 ns per found segment, with one additional clock cycle needed for initial synchronization. These lists of found track segments are stored synchronously in a FIFO memory.

The segment collector then transfers the segment list to the linking subsystem as quickly as possible. Because the FEB crates are located in many different locations around the detector, and the segment lists from all superlayers in all sectors must be brought to one place, the list needs to be transferred up to 100 ft. To minimize the cabling problems involved in transferring this data to the linking subsystem, we are planning to use a Fiber Channel compliant chip set. The Cypress CY7B923 accepts bytes of data at rates from 24-31 MHz, and serializes the data using an 8B/10B encoding scheme. The output bit rate is then 240-310 Mbit/s, and can drive coaxial cable or, with an interface chip, fiber optic cable. We plan to run this communication link at 25 MHz using a high quality low-loss coaxial cable (Belden 9913), to transfer the data from the segment collector to the linking subsystem.

The timing for transfer to the segment linker is as follows. The daughter modules on each FEB produce segment data within 150 ns. Assuming 3 found segments per superlayer (an overestimate) it will take the segment collector another 160 ns to encode and store them in the FIFO. Transfer of the data to the linking subsystem takes another 3 clock cycles plus the propagation delay of the cable or 120 ns + 120 ns, for a total of 550 ns.

B. Segment Linking

Each sector's segment linking subsystem (all operate in parallel) will use the Cypress CY7B933 (the companion chip to the transmitter) to receive the segment list and store it in a FIFO. When the segment lists from all three axial superlayers have been transferred, the linking subsystem begins operation. Tracks will be found using an array of associative memory chips with independent column matching. Each associative memory chip will have three inputs of eight bits, and 256 memory locations. Each memory location is filled with the addresses of the three segments (one in each superlayer) which link up to form a valid track; this combination is called a road. Enough memory chips are used to hold all necessary roads.

The linking subsystem reads the segment list from all three superlayers (each residing in its own FIFO) into the three column inputs of all chips; any memory location which matches the incoming data in its column sets a 'match bit' corresponding to that location and column. When all segments from all three superlayers have been sent through the memories, any memory location with match bits on in all three columns corresponds to a found road. The advantage of this approach is that the time to link all segments in the three superlayers scales as the largest number of segments in one superlayer, and not as the product of the number of segments in each of the three. Thus if there are three track segments in the innermost superlayer and two in each of the others, only three match cycles (as opposed to twelve) will be required to find all roads which match the track segments. A fuller discussion of this approach can be found in reference [6]. A priority encoder within each chip produces a list of found roads (by address); this address plus the chip number are used to find the track parameters in a standard memory lookup. Our simulation results for two cell segments indicate that less than 10,000 different roads need to be stored to handle even extended targets in the CLAS detector.

Because the linking system will be memory based, it will easily handle the reprogramming necessary to handle reduced or reversed magnetic fields, as well as different types of targets. All that is needed is a new simulation run to find the roads corresponding to tracks in the new detector configuration.

V. CONCLUSIONS

The segment finding algorithm for the LEVEL 2 trigger in the CLAS detector has been developed and prototyped in a Xilinx XC-4005 FPGA. A total of 768 of these chips will be used to simultaneously find all track segments in the axial

superlayers of the detector in well under 150 ns. The algorithm is extremely efficient, and finds segments even with two layers missing from the track. It is also highly resistant to finding false segments due to noise.

Track segments will be collected and passed over a high speed serial link to the linking subsystem in an additional 400 ns. The two-cell granularity of the segments will allow the linking subsystem to obtain a resolution of about 20 % in the forward direction.

VI. ACKNOWLEDGMENTS

The authors wish to acknowledge D. Heddle for using his display package to assist in algorithm development, and B. Niczyporuk for generating detector events and performing the segment linking simulations. We are also grateful to Rich Tokosh of Marshall Electronics for assistance in the Xilinx development. This work was supported in part by Department of Energy Contract DE-AC05-84ER40150.

VII. REFERENCES

- [1] Conceptual Design Report, CEBAF Basic Experimental Equipment, CEBAF, April 13, 1990.
- [2] D. Doughty et al "A VXIbus Based Trigger for the CLAS detector at CEBAF." IEEE Transactions on Nuclear Science, NS 39, 1992 p 241-247.
- [3] B. Niczyporuk "Standard Data Analysis (SDA) Package", unpublished..
- [4] D. Doughty and F. Barbosa "Triggering and Acquisition Problems in the CLAS Detector at CEBAF." Proceedings of the Second International Conference on Electronics for Future Colliders, 1992 p 73-86.
- [5] The XC 4000 Data Book, Xilinx, 1991.
- [6] M. Dell'orso and L. Ristori "VLSI Structures for Track Finding" Nuclear Instruments and Methods, A278, 1989 p 436-440.