E877 Note # PITT-95-002

February 8, 1995

Trigger Development in Experiment E877

W.E. Cleland, M. Clemen, U. Sonnadara and S. Voloshin

University of Pittsburgh, Pittsburgh, PA 15260

1 Introduction

Over the last few years, the central collision program of the E814/877 collaboration was heavily involved in addressing the physics topics such as low p_t enhancement, antiproton production and kaon interferometry. A major problem in this kind of analysis is to obtain enough statistics for each type of particles while keeping the background as low as possible. In the June 1990 and July 1991 runs, a trigger in which the time of flight of downstream particles which was at least 3 ns longer than that of velocity of light particles was used to enhance the data sample to study antiprotons. With this trigger, we were able to extract a few hundred antiprotons, after background subtraction. However, this was done using the time of flight from our forward scintillator wall, which has a granularity too coarse for Au-initiated collisions and is located too far downstream for efficient kaon detection. It is therefore not useful as a trigger in E877.

During the 1993 Au run, we installed four multi-wire proportional chambers (MWPC) each with 0.5 cm wire spacing to improve the tracking efficiency in our experiment. This was necessary since the pattern recognition problems for the two tracking chambers (DCII and DCIII) became intractable when 10-15 particles pass through the chambers simultaneously. In addition, a new hodoscope (TOFU) with slat widths between 1 and 2 cm and good timing properties was installed in the upstream region of the experiment to improve the measurement of time of flight of the particles. New phototubes with excellent timing were also installed in the beam scintillation counters to improve the measurement of the start time. We achieved roughly 80 ps resolution in the time of flight with this new system. In order to improve the efficiency for detecting kaons, the new time of flight wall is located much closer to the target than the E814 forward scintillator wall, which increases considerably the difficulty of including this device in a hardware trigger. We therefore proposed a different way to solve the problem in triggering for rare event types. We plan to develop a trigger that can track particles and reconstruct the momentum online. With the aid of the precise timing information, we will make a third level decision on particle identification to decide whether to write the event record to tape.

Since the final stage of the trigger will be carried out in software, the data acquisition system will need to collect data at a higher rate than is possible at present, in order to write data to tape at the current speed (≈ 60 events/spill). As is discussed below, a large factor appears to be achievable if the software for the SSPs is rewritten with speed as the main objective.

In this note we discuss the development of the tracking trigger hardware, preliminary studies of the tracking trigger performance and the enhancement of the data acquisition system software.

2 Hardware Tracking System Developments

2.1 Tests of the Tracking Trigger Electronics

Preliminary tests of the tracking trigger electronics concepts were carried out in the run of 1993. In the fall of 1994, we implemented and tested 1/3 of the full system. This test bore some similarity to the previous test configuration, but improvements were made to improve the precision of track reconstruction and to permit online monitoring, and tests of online time of flight reconstruction were carried out as well.

The instrumented region in the MWPC units we used for the tracking trigger test is shown as the shaded region in Figure 1. Signals from the MWPC units are fed into the PCOS latching modules which store the data for readout through the PCOS controller, which in turn provides the MWPC information for the normal data record. We used the additional "prompt OR" output signals from the PCOS latching units, consisting of an OR of two neighboring wires (this configuration gives us roughly 1 cm readout spacing) as input to the new Latch and Multiplexing units (L&Ms) mounted on a rack near where the beam logic is located. These data are latched with the beam triggers and stored in the L&Ms until a pretrigger followed by a first level trigger occurs. Once the level 1 trigger

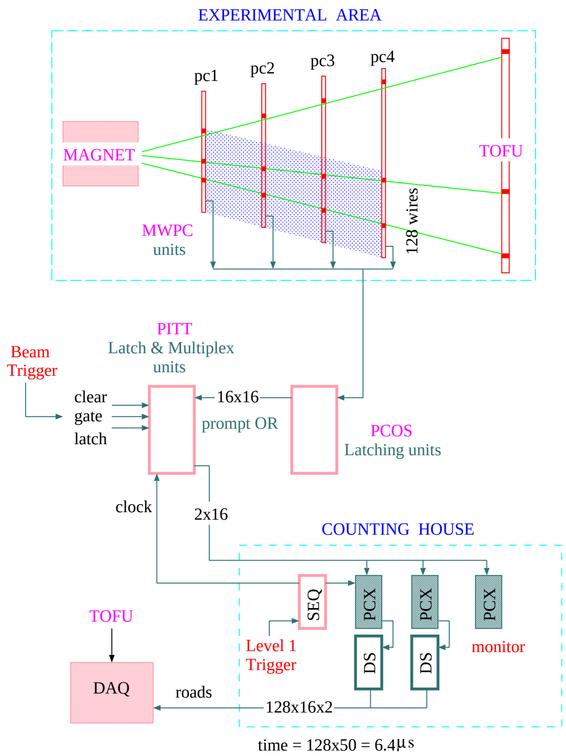

Figure 1: Diagram of the downstream region of the E877 spectrometer showing the MWPC units. The shaded region is the area which was instrumented for the 1994 test of the tracking trigger.

is issued, a signal is sent from the counting house to the L&Ms, which in turn causes them to transmit the MWPC data to the counting house through a serial line (using time multiplexing) to a Programmable CAMAC Xilinx (PCX) module. The general purpose PCX modules, developed at the University of Pittsburgh, are based on the 3000 series Xilinx Field Programmable Gate Arrays and are capable of performing a wide variety of logical operations on two 16-bit input data words and writes the results to a single 16-bit output data word. Introduction of the L&Ms between the PCOS units and the PCX units accomplished two important tasks: (a) intermediate storage of the data until the level 1 trigger is available and (b) orderly transmission of the data into the PCX units.

To generate the possible track patterns to be recognized by the tracking trigger, we used the heavy ion collision simulation code RQMD to generate the tracks in our spectrometer acceptance for the magnetic field to be used for data taking. Each pattern was defined in terms of the offset in the number of readout spacings in the last 3 MWPC units with respect to the first MWPC unit. Due to the large number of possible track patterns and the limitations in the available time to analyze an event, we have grouped all track patterns going through the first and the last MWPC units as one, and defined them as roads. In principle, each road can have a maximum of up to 5 different track patterns since in this geometry, at least 3 out of 4 hits lie in the line formed by the first and the last MWPC units.

The logic in the PCX modules is identical to that used in the 1993 test, where a specific road is identified by coincidence logic, and the translation invariance of the roads is exploited by shifting the bit patterns in the PCX data registers. One major change in the PCX logic used in this test compared to last year is an expansion of the storage registers from 32 to 64 bits. This expansion not only allows the PCX to hold twice as many hits for pattern recognition, but also allows for the recognition of more track patterns, reduces the problems associated with the loss of tracks crossing different regions, and reduces the number of modules needed to store the final results.

In the present configuration, each PCX module is able to recognize up to 32 different road patterns for each of the 64 bit shifts, yielding a total track recognition capability of 2048 patterns. In this test we used 3 PCX modules where 2 were used for pattern recognition and the remaining one was used to monitor the operation of the performance of the PCOS and L&M units. (The monitoring function requires an additional PCX module since in the pattern recognition PCXs, the data are shifted out of the registers and lost.) The output of the pattern recognition results from the pattern recognition modules are stored in two LeCroy Data Stack modules, whose contents form part of the event record. The shift clock runs at 20 MHz, so the track recognition step requires 128 $\times$ 50 ns=6.4 μs. The data stacks and the data registers from the monitoring PCX are read by the data acquisition system and become part of the event record. This feature permits us to study the tracking trigger performance offline as well as online.

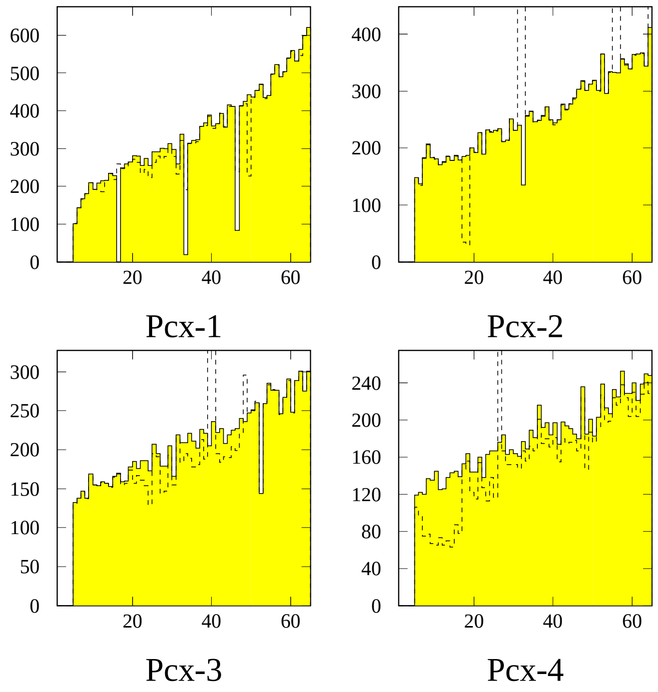

Figure 2: Comparison of the MWPC data from the PCX monitoring module (shaded histogram) with that read out through the PCOS system (dashed lines). Inefficient wires appear as low points in both sets of data. Inconsistencies between the two are attributed to faults in the PCOS system, except for MWPC 1, where two faults in the wiring of the prompt OR signals are evident. The labels refer to the four registers in the CAMAC PCX units, which also correspond to the four MWPC planes.

Figure 2 shows the comparison of the data recorded by the PCOS latching modules (dashed histograms) and the data recorded by the monitoring PCX module (shaded histograms). Spikes in the PCOS data (where PCX data do not show such effects) indicate that the MWPC data are good but the PCOS latching modules artificially generate extra hits. This is quite useful since we can use the data stored in the PCX modules to correct the PCOS data itself, which is used extensively in the offline tracking. Also, the plots indicate that the PCOS data are somewhat more inefficient than the PCX data, especially for MWPC 3 and MWPC 4.

To understand the performance of the tracking trigger, we have implemented the 3rd level trigger logic at the software level. We applied similar corrections and pointers to extract the time of flight corresponding to a given road number where the 3rd level will be used in the real-time tracking trigger operation. In the 3rd level, the time of flight counter number and the momentum are obtained by a lookup table based on the road number, and only the time of flight itself needs to be calculated. Our Monte Carlo studies show that the pointing of the tracks to the time of flight wall is accurate enough to limit any given road to two possible TOFU counters.

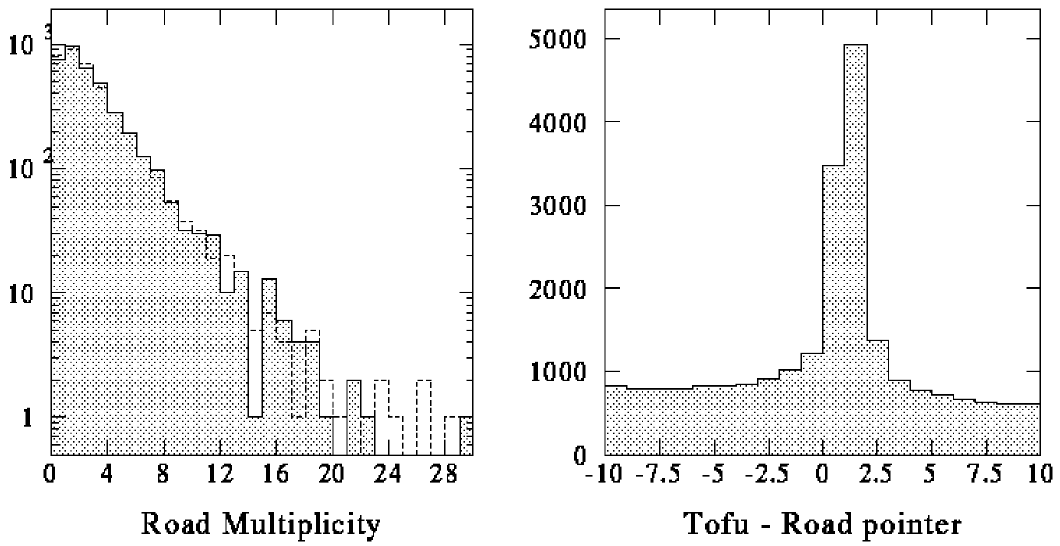

Figure 3: (a) Multiplicity of tracks found by the tracking hardware in the 1994 test. (b) correspondence between the track pointers and TOFU counters containing hits.

Figure 3a shows the observed multiplicity of the roads (solid line) for the region we instrumented. In the same figure we show the multiplicity of the roads with a TOFU coincidence (dashed line). There is no significant reduction using the TOFU coincidence, indicating that most of the tracks are real. Figure 3b shows the coincidence between the TOFU hit counter and the pointer expected from the road number. It is clear from this plot that 2 counters are more or less equally probable. Accidental hits generate a background of about 20%, which are responsible for the flat distribution in this plot.

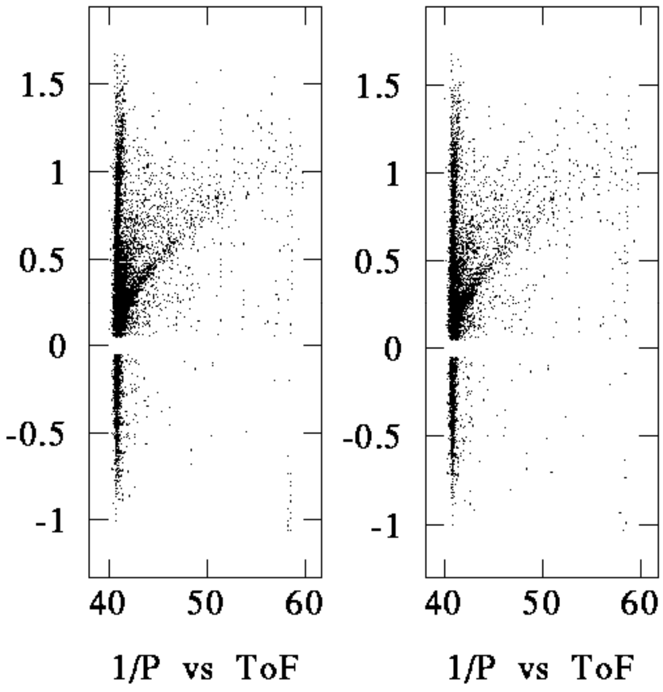

Figure 4: Plot of $1/p$ *vs.* time of flight, which is used for the identification of particles. Left: data produced by the tracking trigger hardware. Right: data produced by offline track reconstruction which simulates the hardware.

Figure 4a shows the particle ID ($1/P$ *vs.* time of flight) scattergram found from the online hardware tracking. In this plot we can identify mass bands for protons and pions. However, kaons are covered by the false tracks. Figure 4b shows the same plot generated offline by simulating the hardware logic. Very good agreement is seen, indicating the PCX track-finding logic was operating as expected.

2.2 The Full Tracking System

To understand the performance of the full system, we used Monte Carlo simulations and real data taken in 1994. Our MC uses events generated by RQMD and creates the hit pattern in the MWPC and the time of flight counters. We have assumed 100 ps for the time of flight resolution, a value which can be achieved at in the 3rd level trigger. We define regions in the MWPC units which will be covered by the different PCX modules and combine the time of flight information and the roads reconstructed from MWPC hits, in the same way as will be done in the 3rd level trigger. If 2 or more particles go through the same counter, only the time from the fastest particle was saved. For the data, we applied the PCX logic at the software level on the hits recorded by the MWPC to generate roads and followed the same steps described above.

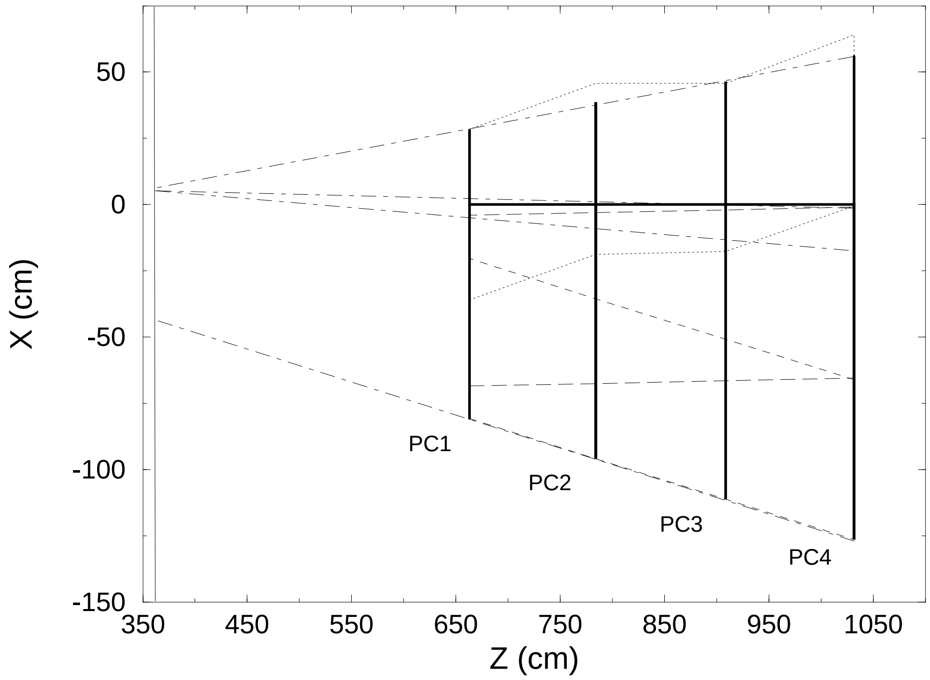

Figure 5: Diagram of the downstream region of the E877 spectrometer showing the multi-wire proportional chambers. The dotted and dashed lines indicate the boundaries of the three regions that will be used by the tracking trigger in the final system. Trajectories for beam rapidity particles for the two polarities of the spectrometer magnet are shown as long-short dashed lines.

Figure 5 shows the number of regions needed to instrument the full tracking system. The region at most negative x (bounded by the short dashed lines) is already implemented as discussed in the previous section. Tracks arriving at shallow angles near the beam line which cross between two regions will not be recognized as tracks. The narrow region where tracks are lost due to this effect is indicated by the short-long dashed lines in the figure. Since this region is very close to the beam region where we do not identify particles even in the offline analysis, the loss in efficiency is minimal.

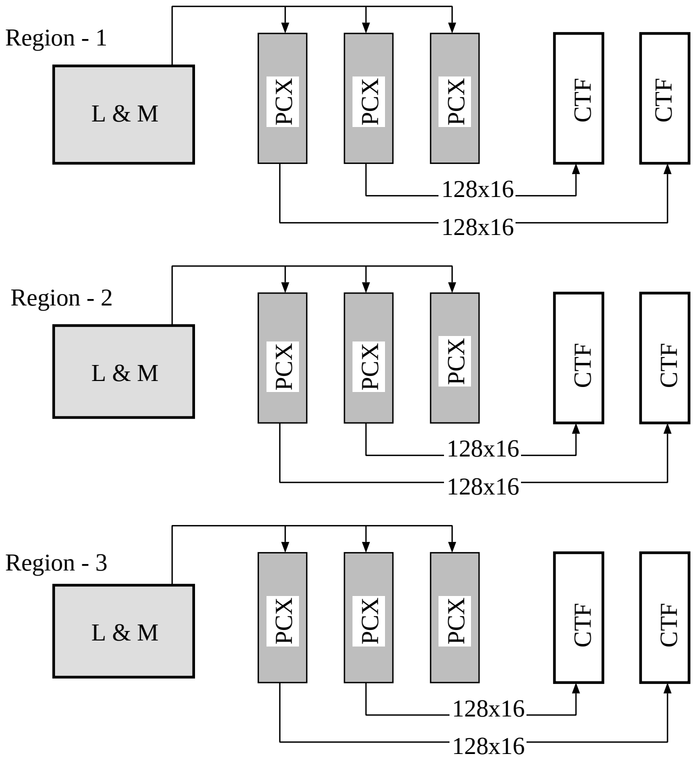

Figure 6: Diagram of the electronics of the full tracking trigger system.

Figure 6 summarizes the electronics needed for the final setup with 3 regions. We need 3 L&Ms, 9 PCX modules (6 for pattern recognition and 3 for monitoring) and 6 CAMAC transversal filter (CTF) modules. The CTF module has also been developed at the University of Pittsburgh. It is capable of performing digital arithmetic functions, but in this case we use it as a replacement for the LeCroy Data Stack modules. The CTFs are preferable to the Data Stack modules since (a) they can be programmed to suppress those data words containing only zeros so that at the 3rd level there will be many fewer data words to handle, and (b) the cost of the module is about a factor of two below the LeCroy unit. Prototype units have been built and operated in the required mode successfully. Finally, one Sequencer module, a unit designed and built several years ago for the E814 second level trigger, is used to control the operation of the tracking trigger electronics.

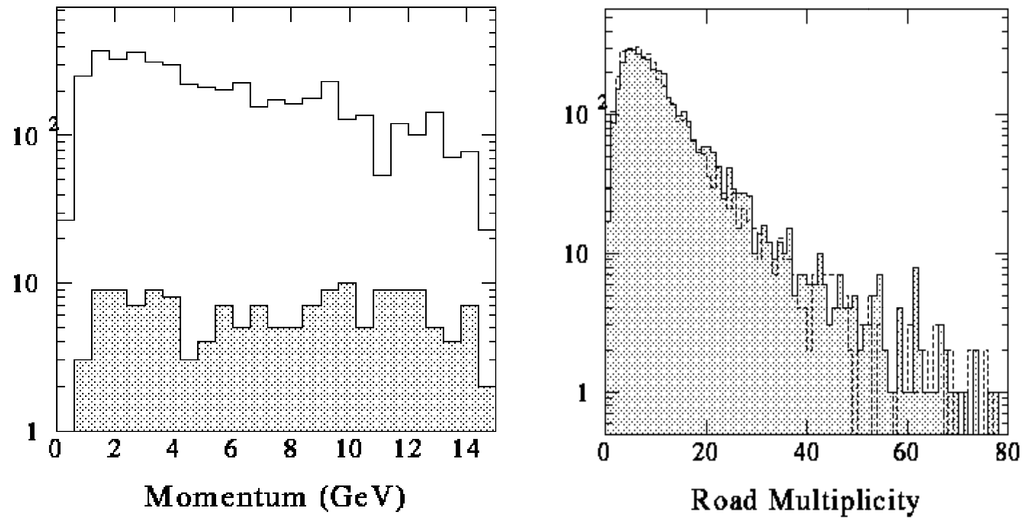

Figure 7: (a) The open histogram shows the expected distribution of successfully reconstructed tracks as a function of their momentum. The shaded histogram shows the corresponding distribution of tracks not found by the tracking hardware. These data are calculated for B polarity of the spectrometer magnet. (b) Solid line: track multiplicity for the full tracking trigger system. Dashed line: track multiplicity when a time of flight counter is required in addition to the track in the MWPC system.

Figure 7a indicates the expected track efficiency for the full system for magnet B polarity as a function of the momentum of the particles. The open histogram shows the reconstructed tracks and the shaded histogram shows the tracks that are missed. Tracks going into the beam region are not considered here. We expect an overall efficiency of above 95% for this reconstruction, but some of the inefficiency comes from not pointing to the correct TOFU counter. This problem is still under investigation.

Figure 7b shows the expected track multiplicity for the full system (solid line) and the track multiplicity with the TOFU coincidence. As was found for the 1994 test data, there is no significant reduction from the TOFU coincidence. One of the possibilities considered in the hardware design was to include the TOFU plane as a fifth element in the hardware track-finding logic. We decided against this option, since it unduly complicated the logic, and this decision is justified by the data shown in Figure 7b. According to these results the third level software will need to perform at least 20-30 TOFU time of flight reconstructions per event on average.

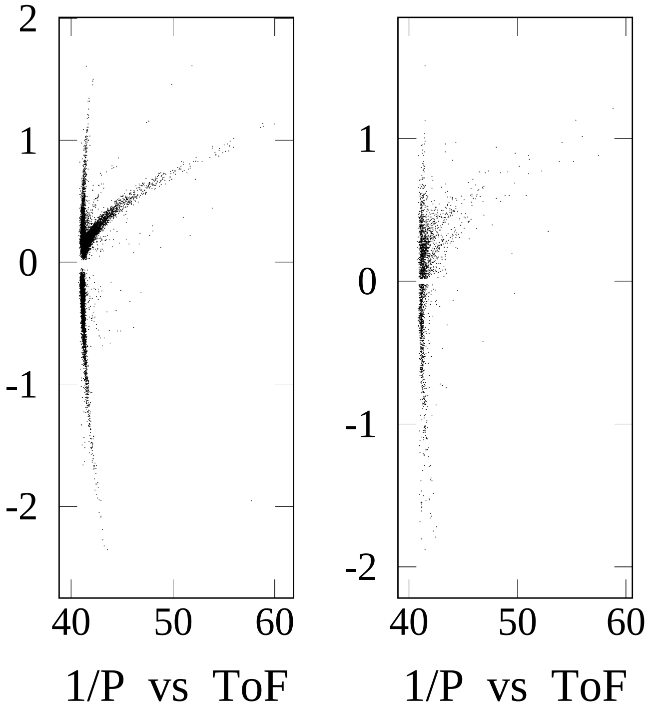

Figure 8: Particle ID plots of the type shown in Figure 4 from the Monte Carlo simulation. On the left are shown the plots for true tracks, while on the right are the corresponding plots for "false" tracks, in which four hits in the MWPCs from more than one track accidentally lie along a straight line are therefore tagged as tracks. Note that there is a certain complementarity between the two plots, showing out the false tracks populate the gaps in the plot on the left.

Figure 8 shows the Monte Carlo generated particle ID plot for true and false tracks. For the true tracks there is clearly enough resolution to identify protons, pions and kaons.

However, the plot of the "false" tracks (which are accidental coincidences of hits of two or more real tracks) are not uniformly distributed but rather form a plot which is complimentary to the one for real tracks. When the plots are superimposed (see Figure 9b), the resulting distribution *appears* to have poor resolution, but for real tracks the true resolution is still that shown in Figure 8a.

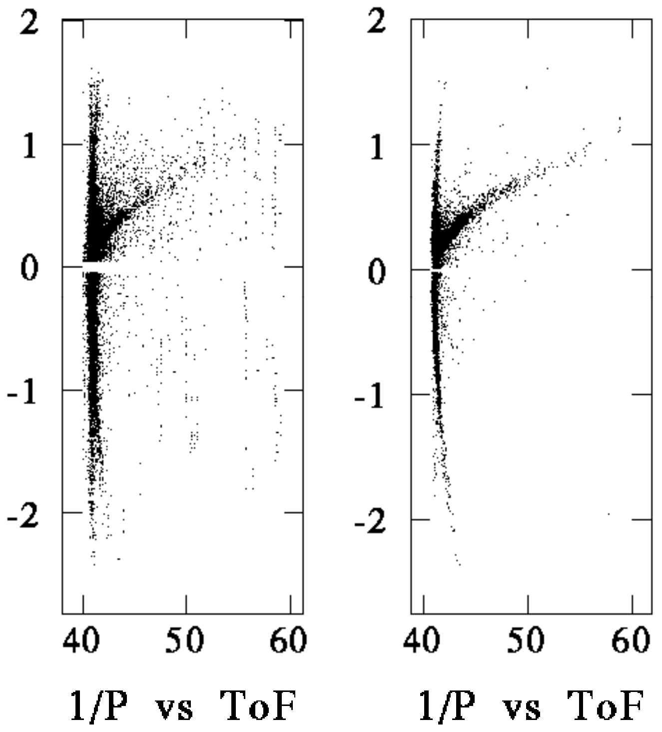

Figure 9: Particle ID plots for the final system arising from reconstructed data in software which simulates the tracking trigger hardware (left hand plot) and from the Monte Carlo (right hand plot). The Monte Carlo plot is the sum of the two plots shown in Figure 8.

Figure 9a shows the same plot for real data, which looks somewhat worse than the MC. One reason for this is the existence of multiple tracks for a given time of flight counter (they are responsible for the vertical lines in the figure), which can be generated by noise in the counter.

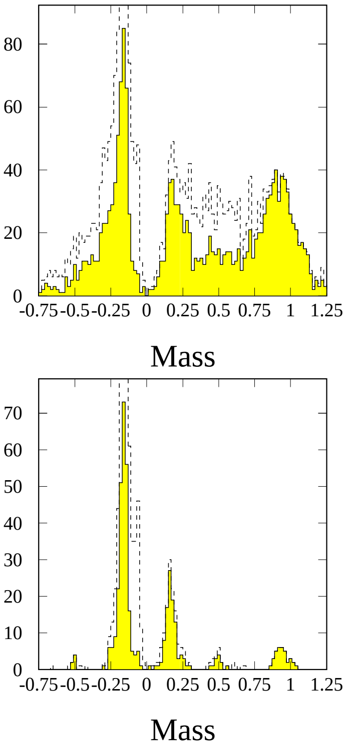

Figure 10: Monte Carlo mass distributions simulated for the third level trigger. In both plots, the open histogram is the distribution for all tracks and the shaded histogram is for true tracks only. In the upper plot, the momentum is required to be less than 7 GeV/c, while in the lower plot, it is required to be less than 2 GeV/c.

Figure 10 shows the reconstructed mass for the Monte Carlo data for 2 momentum cuts. Each figure show mass reconstruction for all tracks (open histogram) and only true tracks (shaded histogram). They show that although false tracks create a significant background, we can nevertheless apply trigger cuts on the mass since we have a high efficiency in selecting true tracks. For the low momentum region false tracks do not have an effect.

2.3 Developments in the offline tracking software

To understand the efficiency in tracking trigger and to correct the triggered for inefficiencies we need a offline tracking algorithm to track particles with the following properties.

- fast execution
- an implementation which is easy to modify
- good performance in high multiplicity events
- well understood inefficiencies
- tracking logic as close as possible to hardware

These requirements are only partially satisified by the existing offline E877 tracking code. We have therefore developed a new tracking algorithm to track particles offline which better satisfies these requirements while perhaps performing worse in the area of track resolution.

The preliminary selection was carried out by searching the possible hit candidates that are generated by the tracks traversing the beam vertex detector, the drift chambers DC2 and DC3, MWPC units and TOFU counters. We used the existing low level unpacking routines for this purpose without any major modifications. The pattern recognition starts by looking for hits in the MWPC units which are consistent with straight lines pointing to the magnet exit. We require at least 3 MWPC units to record hits to be a valid link. We extend the identified links into the drift sections of DC2 and DC3, and the best hits consistent with the given pointers were selected. The region of the search was limited to the pointing accuracy of the MWPC units (at this point, the angle of the tracks are very well defined). For a valid track, at least 4 out of 6 planes from each chamber is required. To select tracks falling very close to each other, pattern recognition is carried out in the same region in the drift sections by removing the already recognized hits. We get about 300 μm for DC2 and 500 μm for DC3 resolution from this method with the 1994 data.

There seems to be some deterioration in the drift chamber resolution, since using the 1993 data with the same technique we find somewhat better resolution (specially for DC3).

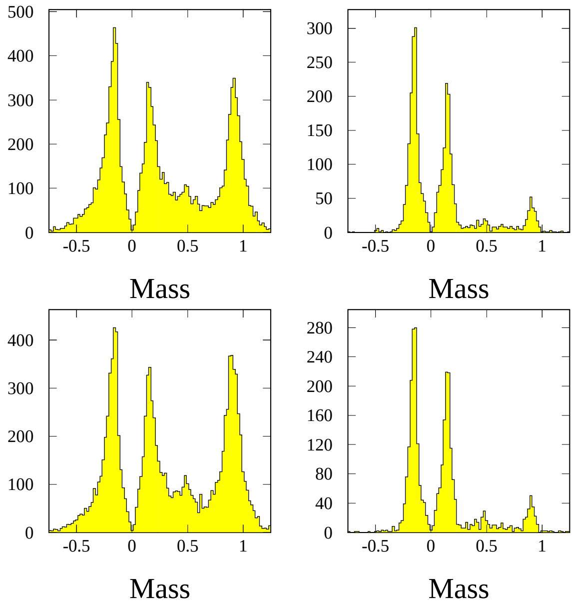

Figure 11: Mass resolution found using offline tracking code. The distributions are for momentum less than 7 GeV/c (left) and less than 2 GeV/c (right). Upper plots: results obtained with the new tracking algorithm described in the text; lower plots: same distributions using the standard tracking algorithm.

The track candidates were further checked for their consistency in the vertical positions extracted by the pad sections and required both DC2 and DC3 pads hits were found. Also, it was required that they contain valid time of flight information at the TOFU counters. The tracks with the same time of flight information and at the same time the shared hits either in DC2 or DC3 were assumed to be a single track and only the best track was selected from such groups. We achieved a factor of 4 improvement in speed and similar tracking resolution compared to the standard tracking code at this point. A comparison between the mass resolution between these two methods is shown in Figure 11. In this plot we compare the mass resolution for the two momentum cuts used above using both the algorithm described above and the standard tracking code.

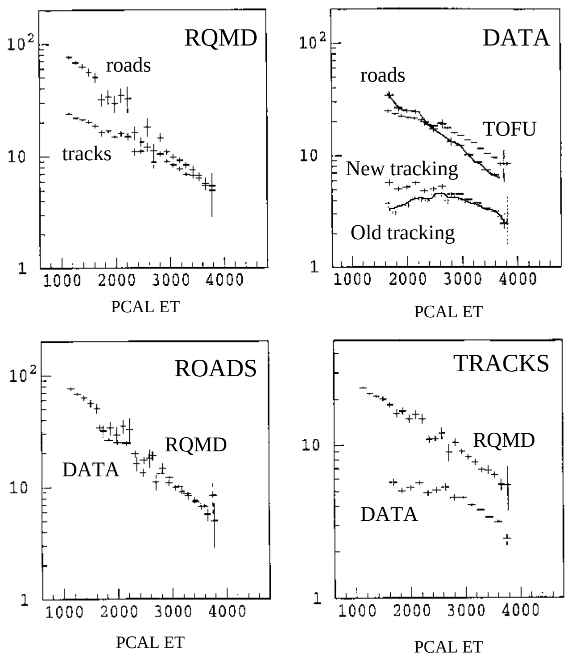

Figure 12: Comparison between the tracks found in the data and the Monte Carlo as a function of PCAL transverse energy (FERA counts). Upper left: roads and reconstructed tracks found from the Monte Carlo; upper right: roads and tracks found in offline tracking, using both the new algorithm and and standard software; lower left: multiplicity of roads from the Monte Carlo compared with the data; lower right: tracks from the Monte Carlo compared with the tracks found from the new offline algorithm.

To understand the efficiencies in the offline tracking, we used RQMD events and subjected them to the same acceptance cuts as the data. Figure 12a shows the track multiplicity as a function of the PCAL transverse energy (FERA counts). In the same plot we show the road multiplicity (which was obtained by applying the hardware tracking logic to the MC events in the software). Clearly, for high impact parameters, when the track multiplicity is high, we produce a large number of roads possibly arising from false combinations. At low impact parameters road multiplicity and the track multiplicity becomes basically the same.

In figure 12b we show the road multiplicity extracted from the 1994 data (upper line) compared with the track multiplicity obtained by our work (labelled "New tracking") and with the previous tracking code (lower line). For high impact parameter collisions, the code we developed has high efficiency compared to the previous tracking code. However, both show low efficiencies compared to the multiplicity extracted from the TOFU counters (labelled "TOFU").

In figure 12c we show the road multiplicity extracted from the 1994 data and the road multiplicity extracted from the MC. They agree very well. This could be due to either

(a) Monte Carlo and the data really have the correct number of tracks and the data does not contain accidental hits in the MWPC

(b) the data has fewer tracks than MC but due to accidental hits the road multiplicity is artificially high and agreement is just a coincidence.

We believe that the answer to this probably lies in between the above two extremes.

In Figure 12d we show the track multiplicity extracted from the 1994 data to the track multiplicity extracted from the MC. The data shows about a factor of 2 less efficiency compared to the MC. We are currently investigating this problem.

3 Studies of Trigger Performance

3.1 Trigger efficiency and trigger rate

A well designed trigger should have high efficiency in selection of desired events along with a relatively low trigger rate. The anticipated total level 3 trigger rate reduction (for a few triggers in parallel) in our case close to a factor of 1/3. This value is limited by the speed of the DAQ and the maximum beam intensity.

Using RQMD generated events we have checked the effectiveness of tracking trigger in selection of the events with positive and/or negative kaons. For this purpose the effective "corridor" on the particle ID plot (1/p vs TOF) was determined. We parameterize the width of the "corridor" with some parameter (which varies from 0 to 1) and calculate the trigger rate and efficiency as a function of this parameter for different magnetic field polarities. We calculate the efficiency with respect to kaons of all momenta and to those with momentum less than 4 GeV/c, below which the resolution of the spectrometer permits separation of kaons from pions in the offline analysis. The results are presented in Fig. 13. From this plot we conclude that the effectiveness of tracking trigger with respect to selection of events with K-mesons is very good. For example one can see that for the trigger condition parameter around 0.3, the trigger rate is close to 1/3. At the same time the trigger efficiency with respect to all kaons is close to 0.85 and with respect to low momentum kaons is closed to 0.95.

3.2 Online tracking trigger constants

The Tracking Trigger philosophy of using TOFU information is slightly different from that of the offline analysis. Trigger tracking and particle identification should be very fast and relatively simple but still with very good timing resolution. A systematic study of different corrections usually applied offline was carried out to understand which corrections are necessary and in which way they should be applied online.

It was found that the knowledge of exact t_0 constants are very important for the online tracking, and the procedure of using them should be different. It is possible and much more convenient to calculate t_0 constants assuming the same path length to each counter (*i.e.*, any physical path length differences are included in the values of the constants). These constants were calculated by the following method. Using the PID plot the pion "corridor" was defined. Then t_0 was evaluated from the formula:

$$t_0 = \frac{pt}{\sqrt{m^2 + p^2}}, \tag{1}$$

in which p is the particle momentum and t is the measured time of flight. In Fig. 14 we present the result of calculation for the run # 7439. One can clearly see the difference

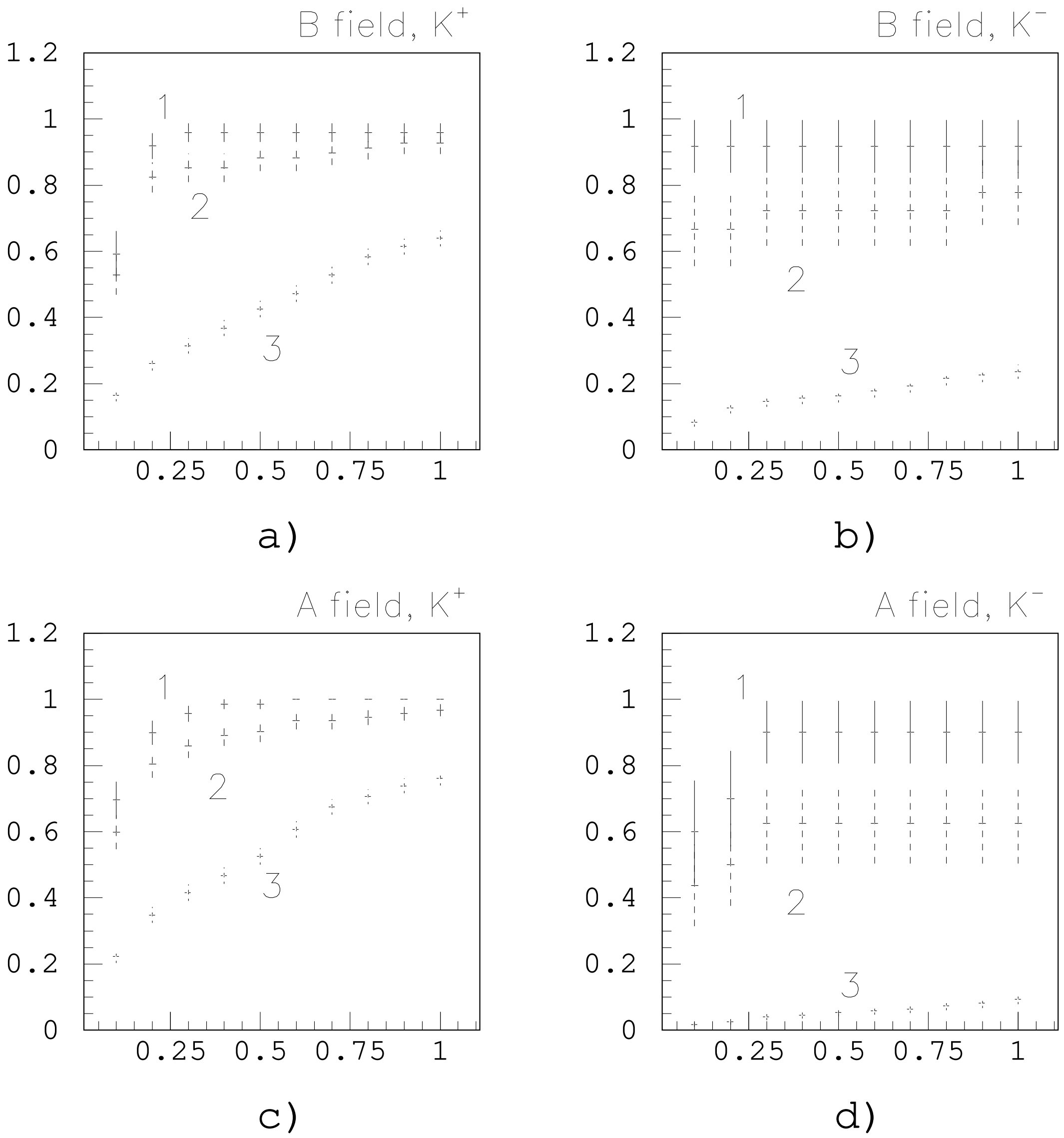

Figure 13: Trigger efficiency for for kaons with momentum less than 4 GeV (1), all kaons (2), and trigger rate (3). The larger error bars for k^-'s are due to the limited statistics of the k^-'s in the montecarlo events.

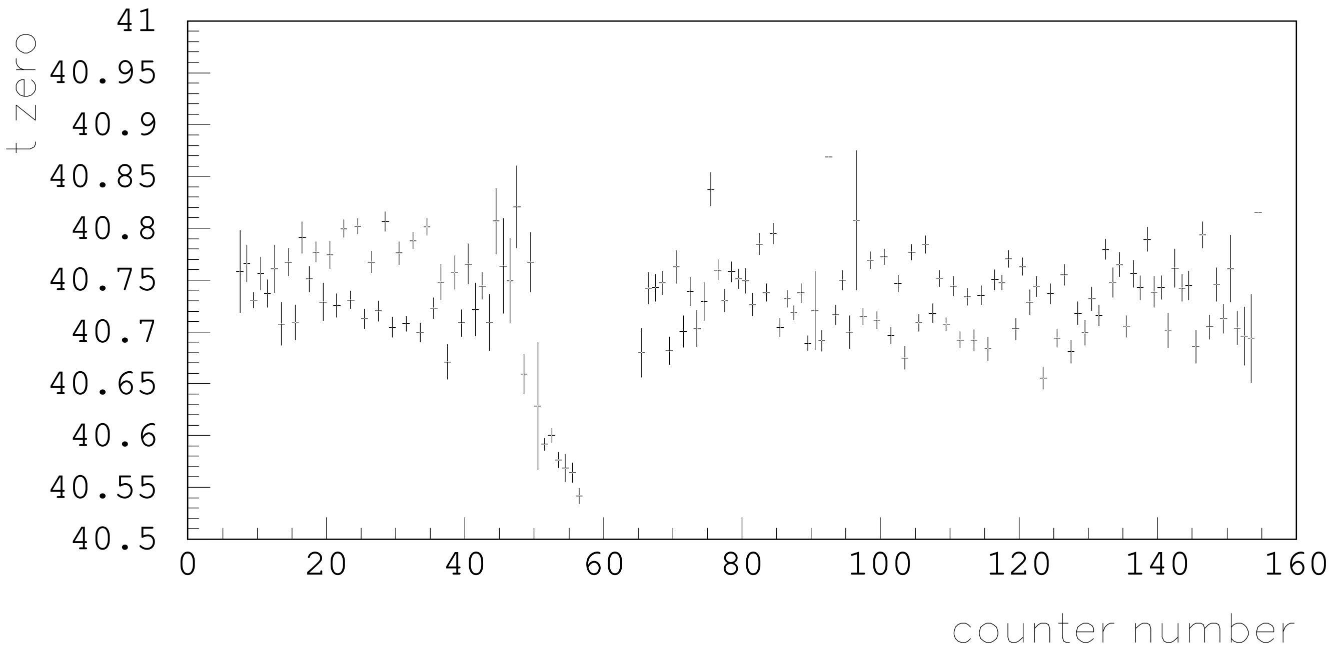

Figure 14: t_0, run # 7439.

between even and odd counters due to space offset about 1.5 cm, which corresponds to a value of about 50 ps.

In the analysis it was found that the corrections related to different path lengths of particles hitting the same TOFU counter but with different momentum are very small (a few ps) and could be omitted in the online analysis. In this case t_0 was calculated by the formula

$$t_0 = \frac{pt}{\sqrt{m^2 + p^2}(1 + \Delta L/L)}, \tag{2}$$

where the relative change in the path length was estimated in the small angle approximation

$$\frac{\Delta L}{L} = \frac{d_1 d_2 (\theta_1 - \theta_2)^2}{2(d_1 + d_2)^2}. \tag{3}$$

Here d_1 is the distance from the target to the center of the magnet, and d_2 is the distance from the center of the magnet to the TOFU.

To define the effective particle "corridors" in PID plot we have to estimate the momentum and time of flight resolution in different parts of the plot. For an estimate of the momentum resolution we found very useful the formula (derived in small angle approximation; the geometry parameters used are from the 1994 run)

$$p \approx 32.5/(x_2 - 361.8\theta_2), \tag{4}$$

where p is the momentum, x_2 is the position just after the magnet (measured in cm), and θ_2 is the downstream angle. The uncertainty $\Delta x_2 \approx 1.5$ cm (which is about the maximum for two wire Tracking Trigger configuration) gives $\Delta(1/p) \approx 0.05\ GeV^{-1}$. We conclude that the momentum resolution in this configuration is sufficient for particle identification. The same formula can be used to estimate the momentum uncertainty due to an off-center beam position at the target, what effectively changes the x_2 value. The beam position at the target varies by only a few mm, so we can safely omit this correction.

The resolution in TOFU can be estimated from the t_0 measurements. It was found that the overall resolution for fast particles ($t - t_0 < 1$ ns) is about 90 ps. But for slow particles (which are mostly protons) the resolution deteriorates in an unexpected way. This problem remains to be understood, but it is not a serious problem for the tracking trigger, since the "corridor" can be made large enough to have good efficiency without a serious increase in trigger rate.

4 Data Acquisition System Developments

4.1 Test Results

Our goal has been to speed up the E877 data acquisition system by at least a factor of 3, and to use a third level trigger running on the VME processor to provide the selectivity against the extra data, maintaining approximately the same tape writing capability. We further are restricted by using as much of the existing hardware as possible.

A histogram of the total data acquisition time, reading out CAMAC only, and CAMAC and 1 FASTBUS crate only in turn, is shown in Figure 15. The total readout time for all crates is 11.8 msec. One sees that the total readout time is dominated by FASTBUS crate 3, although not by a large amount. FASTBUS crates 2 and 4 also show long readout times. The readout time when data compression is not used is shown in the dashed histogramming. One sees that the current data compression algorithm used in the SSP's requires a significant amount of computation time.

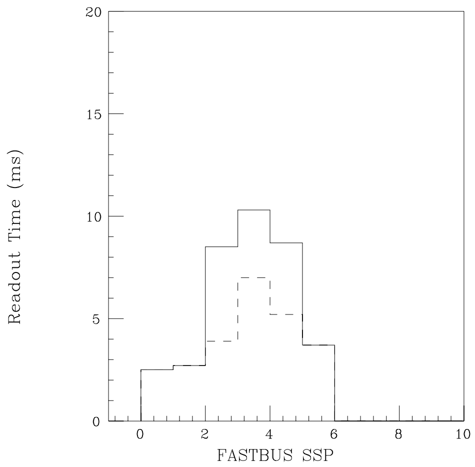

Figure 15: Total DAQ deadtime, reading out CAMAC and one FASTBUS crate in turn. Bin 0 is CAMAC only, other bins are CAMAC and that FASTBUS crate number read out together. The dotted lines represent the DAQ deadtime without data compression.

Our primary goal then implies that the readout time when the trigger is dead must be less than $11.8/3 = 3.9$ msec. This obviously cannot be achieved simply for all FASTBUS crates. We conclude that if one is restricted to the current DAQ hardware, that the low-level SSP software must be re-written, and optimized for speed.

Table 4.1 shows the number of basic instructions to read out each FASTBUS crate, and the total SSP execution time for these instructions, assuming perfect hardware optimization. We assume that DMA readout is available for all of FASTBUS crate 2 - the Drift Chamber ADC's, that single reads must be used for FASTBUS crate 3 - the infamous 1882 ADC's, and that there is some mixture of DMA/single word transfers for the other

FASTBUS crate	Readout Type	SSP cycles	Time (ms)
1	Mixed	3000	0.25
2	DMA	5896	.25
3	Single Read	6240	0.80
4	Mixed	6000	0.5
5	Mixed	3500	0.3

Table 1: Estimated Minimum Readout Times for each FASTBUS crate

crates. The possible readout times are much faster than the current system performance would indicate, and FASTBUS crate 3 is still fast enough with single-word readout.

The time shown in Table 4.1 for FASTBUS crate 2 does not include data compression. It is not clear that data compression is necessary. The 32 kbytes event limit imposed by the magnetic tape buffer could be circumvented by rewriting the recorder so that individual events would span the tape block boundaries. It was found in the last run that leaving out the DC TDCs in the beam trigger, thought to be useless in those triggers anyway, largely obviated the very large size events. The most probable event size then was 16 kbytes. Without data compression in the DC ADC's, this could rise to 21 kbytes.

One sees that the overall data acquisition time is not limited by the TCAL 1882 ADC's, even without DMA, the single read execution time is competitive. Partly this is because the TCAL crate is only 1/2 full, whereas the DC ADC crate is entirely filled. We no longer plan to spend a great deal of manpower forcing the TCAL crate to read out in DMA mode. The foreseen time with single reads is sufficient. This again assumes perfectly optimized software - the software development time would be large.

Table 4.1 shows a possible readout scheme for the 1995 Data Acquisition times. The CAMAC readout time here is left to be large; much of the readout time is in single-reads, and the uncertainty in speedup possibility through the FBD CAMAC branch driver is

DAQ operation	operation time (ms)	Total Time (ms)
Master SSP START	0.05	0.05
CAMAC Readout	2.0	2.05
FB 1 Transfer to Master	0.15	2.20
FB 2	0.25	2.45
FB 3	0.20	2.65
FB 4	0.20	2.85
FB 5	0.15	3.0
Transfer to FVSBI	1.0	4.0

Table 2: Baseline 1995 DAQ operation. This assumes that all FASTBUS readout is completed by the time the CAMAC readout is completed, and that all FASTBUS SSP slaves are ready to transfer their data to the Master SSP.

large. Experience may show that this time can be shortened. Either way, one sees that an overall readout time of approximately 4 msec is reasonably achievable. This though is uncomfortably close to the upper limit of our overall goal.

Data transferred from the Master SSP to the FASTBUS dual-port memory/VSB interface (referred to hereafter as the FVSBI) is then buffered in the FVSBI for readout after the burst, as done previously. The FVSBI has a buffer limit of 1Megx32bits. If we assume an average event size of 21 kbytes, this buffer then is limited to approximately 190 events. Currently we are limited to approximately 60-70 events per spill, for reasonable deadtime. There is still enough memory in the FVSBI to realize out factor of 3 speedup. Again, while the current hardware is sufficient, the restrictions are uncomfortably close to our stated goal.

We have written software for a third-level trigger that operates in the fetcher program, which transfers data from the FVSBI to the VME processor memory once the burst

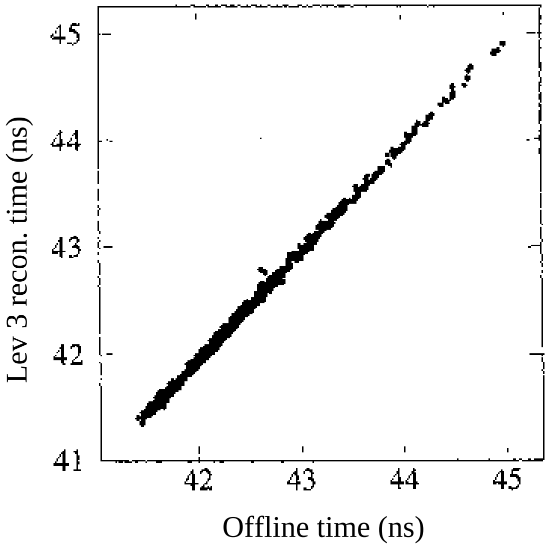

Figure 16: Trigger online TOF versus offline TOF. The vertical scale is the online TOF reconstructed, in nsec. The horizontal scale shows the offline TOF reconstructed. Further improvements in agreement can be made, and are planned.

is over. The accuracy of this third-level trigger depends on the time-of-flight (TOF) reconstruction. Shown in Figure 4.1 is a scatterplot of Trigger TOF to offline TOF. A number of problems were found in the 1994 test program, and much better accuracy is anticipated. More importantly for our immediate concerns, is the execution time of this trigger. Shown in Figure 4.1 is the execution time per TOF reconstruction on the current 68030 processor. The upper points are the times without the floating point co-processor, the lower points the time with the co-processor. One track requires one TOFU reconstruction and one one start time, for a total of 2 TOF reconstructions. The trigger should be dependable for 10-15 tracks, requiring some 15 TOF reconstructions. The total trigger time is estimated under such circumstances to be 48 msec.

Since the third level trigger is supposed to operate on many events (at least 3) for one event written to tape, it needs to operate at 3 times the tape writing speed. Estimates of data acquisition deadtime and total number of events to tape found that the tape drives were operating near to capacity, at 500 kbytes/sec, in 1994. If one event is 16 kbytes, tape

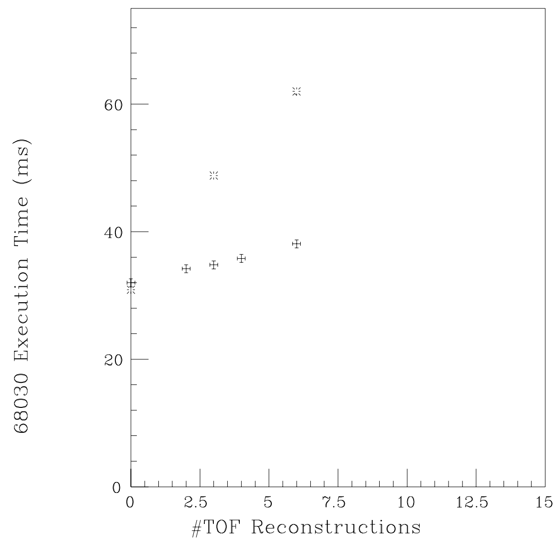

Figure 17: Online Third-Level trigger Execution time, for the computer used in the 1994 running. The stars show the results without a floating point co-processor, the crosses with the co-processor.

writing proceeds at 32 msec per event. The third level trigger then needs to process an event in less than 10-11 msec. The 68030 processor is seen to be completely inadequate to this task.

We have run the same trigger software timing tests on a Motorola 68040 controller at Pittsburgh. The execution time results are shown in Figure 4.1, again without and with the floating point co-processor. Extrapolation to large track numbers gives an event execution time of 10.5 msec. This indicates that a 68040 processor is just sufficient to meet the goals of a factor of 3 speedup. We seem uncomfortably close again to the upper limit of the hardware capability.

This may be mitigated in 1995 by a lower mean track number, which for very central events will be much lower than the number we are using here. Still, we have since spent some time at Pittsburgh working on improving the third level trigger code for speed, and have found that there is significant room for a speedup, as much as 4 msec per event. We conclude that though that the 68040 will be more than sufficient for the operation of the third level trigger in 1995.

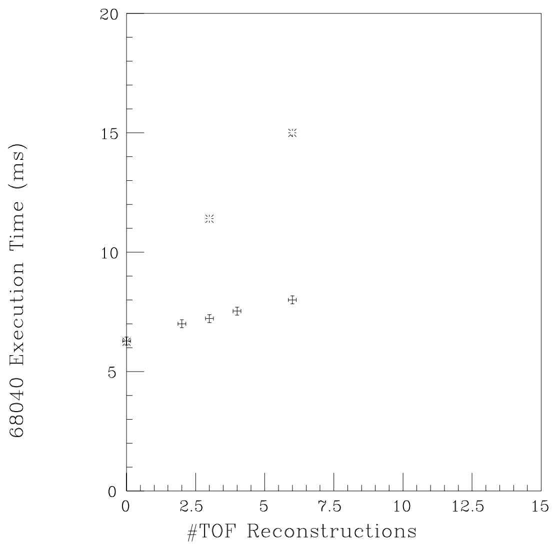

Figure 18: Online Third-Level trigger Execution time, for a 68040 VME-based computer used at Pittsburgh. The stars show the results without a floating point co-processor, the crosses with the co-processor.

4.2 Plans

Table 4.2 illustrates how the DAQ may be operated to push some of the data flow time into the trigger live time. Once a trigger is made, the system SSP begins CAMAC readout, and then polls the FASTBUS SSP's to find which have completed their own readouts. Presumably all have by then. The FASTBUS crate information is transferred to the Master SSP crate-by-crate until all are finished with their individual slave readouts. Again, presumably all have finished by the end of the CAMAC readout. Once all crates are finished their individual readouts, the Master SSP flags the system DONE to the trigger system, making the trigger live to accept more events again. Of course, the Master is not DONE, it must then proceed to transfer the buffered data out of the individual FASTBUS crates into the MASTER, and then to the FVSBI. One sees in Table 4.2 that the time to readout the CAMAC and some of the FASTBUS is approximately equal to the time necessary to read out the rest of the FASTBUS crates and to transfer the data to the FVSBI. For 50% deadtime, this scheme will operate transparently, roughly 1/2 of the DAQ operation actually taking place during the deadtime, and the other 1/2 of the operation taking place during the live time.

DAQ operation	operation time (ms)	Deadtime (ms)	Live time (ms)
Master SSP START	0.05	0.05	
CAMAC Readout	2.0	2.05	
Master SSP signals DONE	0.	2.05	
FB 1 Transfer to Master	0.15		0.15
FB 2	0.25		0.40
FB 3	0.20		0.60
FB 4	0.20		0.80
FB 5	0.15		0.95
Transfer to FVSBI	1.0		1.95

Table 3: Option-A 1995 DAQ operation, pushing data transfer into Live time

We still are faced with a limited amount of memory in the FVSBI. One can further enhance the Data Acquisition by beginning transfers to the VME during the burst itself. As memory is filled during the spill, it is emptied (at a slower rate), so that the total capability for the whole spill is more than the 190 events that we assumed above. We thought of transferring all of the data directly through the FVSBI into VME during the spill, but the FVSBI-VME connection runs 3-5 times slower than the FASTBUS-FVSBI connection creating more deadtime than one would save. By emptying the FVSBI while it is being filled, we have given ourselves some lee-way in the overall amount of FVSBI memory being used. By pushing some of the FVSBI-VME data transfer into the burst, one also allows the third level trigger software to operate during the burst, gaining greater overall time for execution.

There is a penalty for doing data transfers from the FVSBI to the VME during the spill. The FVSBI is capable of doing all of the arbitration necessary. However, the dual-

DAQ operation	operation time (ms)	Deadtime (ms)	Live time (ms)
Master SSP START	0.05	0.05	
CAMAC Readout	2.0	2.05	
FB 1 Transfer to Master	0.15	2.20	
FB 2	0.25	2.45	
Master SSP signals DONE			
FB 3	0.20		0.20
FB 4	0.20		0.40
FB 5	0.15		0.55
FVSBI transfer (1/2 speed)	2.0		2.55

Table 4: Option-B 1995 DAQ operation, pushing data transfer into Live time

port memory does equal-time sharing between the two memory accesses taking place simultaneously. The Master SSP will be able to write to the FVSBI at half the rate as before. This case is shown in the data acquisition scheme in Table 4.2. The overall readout time is larger by approximately 1 msec, although again it could be nicely divided between dead- and live-time operation.

4.3 Near-Term Requirements

The feasibility of the third level tracking trigger and data acquisition speedup has been shown, with some options to further enhance the overall system capability. However, the software development time is expected to be a full-time job from now until the beginning of the next run. We have at Pittsburgh a FASTBUS crate, as well as a VSB/VME crate, a 68040 Motorola controller, and a FASTBUS card for readout testing.

For a system mock-up we have transferred to Pittsburgh 2 SSP's, a Struck FBD

CAMAC branch driver, a CES FVSBI dual port memory and driver, and the cable segment ancillary logic. We have started with the SSP programming using the LeCroy PC-FASTBUS interface, with an 1821 FASTBUS Sequencer and the LeCroy LIFT3 software for low-level system exercising. The full system will eventually run on the 68040 VME system, and will be be transferred back to BNL in the late spring/early summer of 1995, for complete system testing there.

The collaboration has agreed to the purchase of a FIC 68040 controller for E877, which includes more than 8 Mbytes of memory. The current FIC 68030 has been operating with a slim margin at 4 Mbytes. The greater data rate and the lookup tables intended for a more efficient third-level trigger operation will require the more memory. This purchase will be made by Stony Brook in the near future.