The NA48 Charged Trigger : A High Rate, High Precision, Multiprocessor System

Shebli Anvar, Michel Mur

Commissariat à l'Energie Atomique
Département d'Astrophysique, de physique des Particules, de physique Nucléaire et de l'Instrumentation Associée

Abstract : NA48 is a CERN-SPS experiment designed for a fine measurement of CP violation. This paper presents works on the NA48 charged trigger system. The job is to deliver a level 2 trigger for an event rate of about 200 kHz. The triggering involves track reconstruction and mass and transverse momentum calculation. The proposed solution is an asynchronous, parallel, DSP-based machine which will handle floating point computaions in order to deliver, on-line, high precision results at the given rate.

1. The NA48 experiment [1]

NA48 is a fixed target experiment, using the SPS at CERN. The main purpose of the experiment is to make a fine measurement ($2x10^{-4}$) of the CP violation parameter Re(ε'/ε). To reach the proposed precision, the idea is to achieve a simultaneous measurement of the short Kaon and long Kaon decay rates. NA48's data aquisition system is "dead-timeless", in the sense that there'll be no start/stop and detection will be continuous. Amongst other devices, the experimental apparatus will consist of an electromagnetic calorimeter, designed to detect K0 neutral decays and a magnetic spectrometer for the detection of charged decays. An important detector is also the tagging system which is meant to discriminate between KS and KL decays.

2. The magnetic spectrometer

The magnetic spectrometer is made of two sets of two drift chambers separated by a central dipole magnet. Each drift chamber consists of four views, measuring the X, Y, U and V coordinates, the (U,V) frame being the (X,Y) frame rotated by 45 degrees. Each drift chamber view is made of 512 wires. Since the U and V views of the third chamber will not be used, the magnetic spectrometer features 6144 detection channels. Off-line, a resolution of 100µm on coordinate measurement is expected to be achieved.

3. The charged data acquisition system[1]

Since any hit in the chambers is individually stamped at capture time, the data flow is not required to maintain a sychronous time progression of all signals in the system but rather only satisfy average statistical distributions of hits in time and space. NA48's charged data collection system is completely asynchronous because all data carries its detection time with itself.

The detection time consists of a coarse time stamp expressed in units of 25ns, and a fine time with a binning of 1.6ns (1.6=25/16). The time stamp is determined by a general 40MHz global synchronization clock, common to all detectors, while the fine time is measured by local verniers implemented in the wire chambers' TDCs. The collected data carries also a wire number information which, together with the time information, will allow us to ultimately obtain the hit coordinates at a precision of about 300µm on-line. Each view in a chamber is connected to 32 TDCs (one TDC chip for 16 wires). The TDC chips contain an internal FIFO in which they store the digitized data. The 32 TDCs of a view are read in parallel by a device which serializes the data into a final FIFO which, in turn, outputs its data into a circular buffer (called also a "ring"). At this junction, the location at which the data is stored in the ring is

---

[1] See figure at end of paper.

determined by its time stamp. The whole ring may contain 200 μs of event history.

A ring may be read by two devices : the readout system and the level 2 trigger machine (also called "massbox"). A massbox read of the ring is prompted by a level 1 trigger whereas the readout reads the ring only if prompted by a level 2 trigger. If the level 1 trigger system determines that there may be an interesting event around a given time stamp, it triggers the massbox which then reads from the ring the data situated around that time stamp. That data then goes into a coordinate generation device which sends its results to a machine which will achieve vertex, mass and transverse momentum computations. Eventually, the massbox sends its response to the level 2 trigger supervisor. After receiving the responses of all the detectors' trigger systems, the trigger supervisor makes a decision which, if positive, is sent to the different detectors' readout systems. Upon receiving that signal and its associated time stamp, the rings' readout controllers achieve a second read of the aforesaid data and send it to the level 3 system through an optical link.

Since the data path from the TDCs to the readout is completely asynchronous, the system will have to guarantee maximum travel times during which the data will have to flow through each stage of the process. For example, we have to be certain that by the time the massbox reads a ring, all of the relevant data has had enough time to reach the ring ; thus, it is essential that the travel time from TDC to ring be bourned. For the same reasons, the massbox response must be prompted to the trigger supervisor not later than 100 μs after the front end capture of an event.

4. The coordinate generation device

The coordinate generation device is a filter between the rings and the massbox. Its has two functions : first to produce integer coordinates from the raw data that is read in the ring, second to use the redundant U and V coordinates to associate X and Y and to lower the impact of detector inefficencies. The first function will be achieved by a home-made, pipelined, hardware machine. The solution for the XY association is less straightforward : The XYUV association process runs compares the X,Y, and U,V lists and extracts the actual space-point coordinates. This process must find points which were measured in all projections, but should also accept incomplete (3 planes) combinations which could result from detection inefficiencies in the chamber.

Various implementations targeted to DSP microprocessors, and to the ASPEN parallel machine are considered. The ASPEN (ASP embedded node) architecture associates a DSP microprocessor and the SIMD ASP processor into a single scalar and vector machine [2].

A naive sequential version of this matching algorithm must develop "compute and match" association loops to compare the expectations, reconstructed from the measurements in two projections, with the actual measurements in the other two projections. This computation is achieved in several runs to ensure that all potential points with only 3 measurements were found, while authorizing actual solutions where several points project to the same location in one of the X,Y,U or V views. Such algorithms use multiple nested loops and exhibit a high dependancy on track multiplicity (n^3 dependancy).

The sequential association algorithm may be improved by the construction (and later destruction) of intermediate content-addressable look-up tables. If for example the U vector has to be checked against the expectations derived from the X and Y vectors, a U Look Up Table is first prepared. This LUT is filled with "presence flag" words at locations derived from the U values. Neighbour locations are also filled to accomodate later inexact interval matches. Then, the list of n^2 expectations for U is built from the (n-valued) X and Y vectors. Each expectation value is used as an address in the U LUT, and the actual presence of a U flag information at this location is checked. The resulting algorithm exhibits a n^2 dependancy, with additional overhead related to the construction and deletion of the table emulating the content addressable memory.

The XYUV algorithm was also simulated on the SIMD kind Associative String Processor architecture, used in two different modes :

SIMD in place calculation : This mode implements the nested loop algorithm. It first installs all (n^2) XY combinations in different Processor Elements by two successive loads for the X and Y vectors. This operation uses the broadcast capability of the ASP, and performs the distribution of all XiYj pairs in O(2n) time. Then, parallel, in place computation is performed to compute the Uij expected value in each PEij Processing Element (constant time). The actual U measurement vector is then proposed simultaneously to each PEij target for association (O(n) time). The array is then read back to obtain the indexes of the matching XY pairs (O(n) time).

Geometric look-up table : This mode uses the ASP array as a geometric image of the detector. The chamber surface is binned in X, Y, U, and V intervals. Each resulting region is allocated to a single PE, which is pre-loaded with its own xyuv position. For each event, the X,Y,U, and V measured vectors are successively proposed to the array for association. During this operation, the relevant PEs locally store the match information for each projection, and the corresponding vector index (O(4n) time). The array is then read back to obtain the indexes of the associated XYUV values from the matching PEs.

5. The massbox

Ideally, we are interested only in two-tracks events, since the purpose of the drift chamber system is to study the $K \to \Pi^+\Pi^-$ decay. But due to accidentals and other noises, we will probably have to consider also three-particles events. The massbox is thus being designed with the capability of handling configurations with up to three hits in each view.

The vertex and mass computations for a "clean" two-particles configuration take about 2.5 µs on a high-performance Motorola 96002 DSP : if we had to work only on clean cases, a single DSP would make the job. The problems arise when we consider the three-tracks events, essentially because of combinatorial complications. For three particles, a single DSP would have to make 9 times more computations (one for each possible pair of tracks), in which case, vertex amd mass calculations would take a hefty 23 µs, not to mention the overhead due to loop management.

To ease the toll taken by combinatorials, the natural idea is to distribute the computations to a farm of DSPs working in parallel. The main problem then is how to dispatch the data between the DSPs and how to collect the results afterwards. The latest Texas Instruments' TMS320C40 DSP appears to be a possible solution to our problem : it has roughly the same computational power as the 96002, two 32-bits data busses, and *six asynchronous communication channels* which can transfer data at a rate of 20 Mbytes/s. These channels make it very easy to build any network of communicating DSPs. In our case, the most straightforward idea is to have a master DSP (called the "combinatorials' dispatcher") which prepares the different possible two-tracks events and then distributes them to a farm of "workers" through its channels. The workers would then send their results to a "collector" DSP which would produce the massbox's answer and pass it to the trigger supervisor. Presently, the architecture of the DSP network is still to be defined. We may however foresee that it will be spindle-shaped and will roughly contain a dozen DSPs. It will be fully reprogrammable and, consequently, improvable : of the reasons which made us opt for a DSP-based solution, flexibility is not the least. We have begun tests on a prototype network of four TMS320C40s to study different routing and dispatching algorithms, and to have more precise measurements of the timings, especially the timing overheads due to inter-processor communications.

We expect the event rate with which the massbox will have to cope to be of 200 kHz. This yields an event's average computing time of 5 µs. The massbox machine we have just tried to feature will clearly need more, probably around 20 µs. Thus, to be able to cope with the throughput, we must introduce a second level of parallelism : we'll need something like half a dozen DSP machines working in parallel in a crate and managed by an "event dispatcher". The whole massbox apparatus will thus be scalable in the sense that we'll be able to reduce dead time by adding new modules in the crate.

[1] "Proposal for a Precision Measurement of ε'/ε in CP Violating K^0->2Π decays", CERN/SPSC/90-22 SPSC/P253, July 20 1990.

[2] H. Le Provost, M. Mur, B. Thooris, "A real time environment for the Associative String Processor in second level trigger applications", Seventh Conference Real Time '91, p 199.

NA48's Charged Data Collection and Trigger