FERMILAB-CONF-1984-141

A PARALLEL, PIPELINED, EVENT PROCESSOR FOR FERMILAB EXPERIMENT 605

D.M.Kaplan
Fermilab

ABSTRACT

I describe the E605 event processor, which represents the first use of the Nevis Laboratories processing system[1] in an experiment. The processor is constructed of simple general-purpose modules designed to operate on 16-bit data words synchronously and in parallel at speeds up to 40 MHz. It uses information from hodoscopes and wire chambers to reconstruct tracks in the bend view of a magnetic spectrometer, calculating an approximation to transverse momentum (p_t) for each track and mass for the most massive pair in an event. It also distinguishes tracks originating in the target from tracks originating in the beam dump. The results from the processor may be used to prescale events of low mass or p_t, and to simplify further analysis of the events by an on-line array processor or off-line computer. The processor was debugged during the Winter run of the Tevatron and is being utilized in the Spring run.

1 Introduction

Fermilab Experiment 605 is a spectrometer for the study of particles and pairs of particles produced at high transverse momentum (see Figure 1). Its open geometry permits simultaneous detection of leptons and hadrons, and its high degree of segmentation, sophisticated triggering scheme, and high-speed data acquisition system are designed to allow data taking at the very high beam intensities required to study phenomena representing 10^{-11} or less of the total proton-nucleon cross section. After a test run in the Spring of 1982 and a 400 GeV data run in the Fall of 1983, we are now taking data at 800 GeV and 10^{10} proton interactions per accelerator pulse (10^{9} interactions per second).

To enhance the power and flexibility of our triggering scheme we have constructed a parallel pipelined event processor to reconstruct particle tracks on-line. Using the granularity of the wire chambers, this processor is able to make more precise decisions than the first- and second-level triggers, which are based on hodoscope and calorimeter information, and it is faster and more powerful, though more specialized and less flexible, than the array processor attached to the on-line computer, which constitutes the final level of on-line filtering[2]. In addition to reconstructing tracks, the processor computes an approximation to transverse momentum (p_t) for each track and mass for the most

FERMILAB-CONF-1984-141

massive pair of tracks in each event. These can be used to reject or prescale events of low p_t or mass.

Originally the processor was conceived as a means of rejecting particles originating in the beam dump rather than in the target, but our goal has evolved into the distinguishing of target events from dump events, with different p_t and mass thresholds applicable in the two cases. This evolution was motivated by our realization that although muon pairs produced in the beam dump have worse mass resolution than those from the target by more than an order of magnitude, they have five times larger acceptance in the spectrometer and are accepted in a different region of phase space (large Feynman x), where interesting physics may be found.

2 General Description of Processing System

The processor is constructed using the Nevis Laboratories data-driven processing system[1], which consists of a set of processing modules and a standardized bus and protocol for interconnecting them. Each module implements some simple operation such as addition of two 16-bit quantities, comparison of a 16-bit quantity with upper and lower limits, or computing an arbitrary function of an 8-bit quantity via table lookup. The modules used in the E605 processor are listed in Figure 2, and Table 1 summarizes the processor bus protocol.

The modules are designed to operate with a cycle time of 25 ns. It takes typically two or three cycles for an input to propagate through to the output, but the modules are internally pipelined so that a new operation can begin every cycle. Data transfers between modules are also pipelined, so that after an initial period in which the processing pipelines are filling, all modules in the processor are operating simultaneously to the maximum extent possible. Data transfers among modules are synchronized to a central clock, which may be single-cycled, speeded up, slowed down, or run in bursts for diagnostic purposes, and every register and counter of every module may be read or written under computer control, facilitating thorough testing of the system.

Compared to hard-wired special-purpose processors, the system is quite flexible, since the modules may be recabled or the lookup tables reprogrammed to make changes in the algorithm. It also differs from typical hardware processors in that the algorithm is embodied in the interconnection of the modules rather than in control logic, so that no bottleneck arises when the system is expanded; as more modules are added, the total processing power increases proportionately.

The modules are constructed on 9"-square two-layer printed circuit boards. Most modules consist of a single board, containing typically 60 ECL 10,000 chips, and can be duplicated

for approximately $200 (an exception is the Map module, which takes two boards). Each module connects to one or two input processor busses and one output processor bus, as well as to power and control busses. The control bus is a simple bit-serial bus used for initialization, downloading of data, and testing.

3 Reconstruction Algorithm

Reconstruction is performed only in the y-z plane (the plane in which the magnets deflect), using the six wire chambers Y1A, Y1B, Y2, Y2', Y3, and Y3'. Y1A and Y1B are 2-mm-spacing proportional chambers, while Y2-Y2' and Y3-Y3' are pairs of drift chambers with cell widths of 1 and 2 cm (respectively), and with primed chambers offset by half of a cell with respect to unprimed chambers.

For all particles traversing the three stations of chambers, the positions measured by the chambers are linearly related:

$$Y1 = aY2 + bY3 + c \ .$$

The trackfinding loop forms all possible track hypotheses, consisting of a hit in station 2 and a hit in station 3, projects to station 1 using the above relation, and demands a hit in station 1 consistent with a particle originating in the target or in the upstream end of the beam dump.

The present version of the processor uses only wire numbers and ignores drift times, except to cut out hits whose drift times are too small (since with our drift chamber electronics, the first two time bins contain only hits from previous proton interactions). At typical beam intensities, this cut eliminates approximately 30% of drift chamber hits. With ~10 hits per plane, this algorithm is capable of finding all the tracks in an event in typically 5 microseconds or less. The efficiency of the algorithm depends on the efficiencies of the hodoscopes and chambers; since these are all well above 90%, we can expect an efficiency greater than 95%.

4 Detailed Description of the Processor

Figure 3 is a schematic diagram of the processor. The following paragraphs describe the function of each module, proceeding roughly downwards from the top (i.e. in the direction of data flow).

In order to reduce sensitivity of the algorithm to chamber dead time and other inefficiencies (typically 10% per chamber plane), all track hypotheses having a hit in each station plus at least one additional hit are accepted as tracks (i.e. two of the six planes may be missing). This is facilitated by merging the

hit address lists from primed and unprimed drift chamber planes before hypothesis generation, using the Ordered Merge and Associator modules. The Ordered Merge merges the lists while maintaining monotonicity of wire number, and it appends to each address a low-order bit indicating primed or unprimed plane. The Associator finds pairs of adjacent hits in a primed and an unprimed plane and squeezes them into a single address, appending a low-order bit to indicate pair or single hit. Hit addresses emerging from the Associator thus represent position in units of 1/4 wire-spacing.

Between the Ordered Merge and the Associator are placed a Block Buffer and a Comparator. The Comparator eliminates early hits from previous interactions (as mentioned in Section 3) by cutting on drift time. The Block Buffer is a 256-word memory with three ports: a processor write port, a processor read port, and a bidirectional Transport Bus port. (The Transport Bus[3] is Nevis Laboratories' answer to CAMAC and FASTBUS, and serves to interconnect the on-line computer, the processor, and the various pieces of the data acquisition system.) The Block Buffer stores up to 255 hit addresses emerging from the Ordered Merge and sends them to the processor. If the processor accepts the event, the hit addresses are then read out to the on-line computer via the Transport Bus. The Block Buffer is designed to connect subsystems having independent clocks, allowing the processor to cycle at a higher speed than the Transport Bus or readout system.

4.1 Hodoscope Masking

The three modules following the Associator use hodoscope information to reduce the number of drift chamber hits to be used in hypothesis generation. Since the hodoscopes are sensitive to a much narrower time window than are the drift chambers, many particles from previous or subsequent interactions register in the drift chambers but not in the hodoscopes; these "out-of-time" drift chamber hits are eliminated by keeping only those hits which correspond to hodoscope hits. The usual computer algorithm for this requires searching a list of hodoscope hits for each drift chamber hit (or vice versa), which can take considerable time if there are many hits. In the processor, this search time is eliminated by using the Map module, which contains an associative memory structure: when presented with an input position, it immediately fetches the state of the counter or wire at that position, along with the state of up to four adjacent counters or wires to either side.

The first Normalizer converts position in the drift chamber into units of hodoscope counter widths; its output is thus the number of a hodoscope counter which should have fired if the drift chamber hit is in-time, as well as some fraction bits indicating where within the counter the particle passed. After all hits from the appropriate Y hodoscope plane have arrived at the Map's write

FERMILAB-CONF-1984-141

port and been stored, the Map can accept input from the Normalizer. It uses the integer part of the Normalizer output to fetch the state of the requested counter plus one counter to either side, and it puts this out along with the input bits. The second Normalizer then makes a decision as to whether the drift chamber hit should be kept, and it puts out the hit address along with a bit indicating its decision. If the drift chamber hit is near an edge of the central counter, the hit is also accepted if the adjacent counter fired, but if the hit is away from an edge then the central counter alone is considered.

(It is evident that the Normalizer is an unusually flexible module, and a few words on its design are in order. The Normalizer consists of two 256-word x 16-bit tables whose outputs are summed. The high-order 8 bits of an input word can be sent to one table and the low-order 8 bits to the other, allowing any function of the form

$$f\ (x_{high\text{-}order}) + g\ (x_{low\text{-}order})$$

to be computed by preloading the tables appropriately. For example, a Normalizer can be loaded to put out its input times a constant. The Normalizer may be divided into "pages," each page to be used in a different case as determined by the input data, by sending some bits from the input to both tables in common. The partitioning of the input bits is determined by a jumper patch programmed by the user.)

Note that the hodoscope data pass through a Normalizer prior to reaching the write ports of the Maps. This Normalizer is used to transform the hodoscope hit addresses into counter numbers, and it also puts out bits indicating plane, which allow each Map to decide which hits to accept. Thus one Map accepts hits from plane Y2, another from Y3, and a third from Y1, and hits from the remaining hodoscope planes are ignored.

4.2 Trackfinding Loop

After being masked with the hodoscopes, the drift chamber wire addresses are stored in the Lists and counted by the Index Generator. The Index Generator puts out index pairs (8 bits per index) which call forth from the Lists all possible track hypotheses consisting of a hit in Y2 with a hit in Y3. The Page Generator and List-Counters generate multiple passes around the loop for each hypothesis. The Page Generator simply repeats each index pair, once for target and once for dump. The List-Counters pass the index pairs through and also store them internally for potential use on subsequent passes. The passes are identified using the Name bits of the processor bus.

On the first pass, the four Normalizers and two Adders (Arithmetic Operators jumpered to perform addition) generate

FERMILAB-CONF-1984-141

predicted positions in the Y1A and Y1B proportional chamber planes, and the Maps and Binary Table accept or reject each track hypothesis, requiring a hit consistent with the prediction in Y1A or Y1B, and a total of at least four out of the six chamber planes. Outputs from the Binary Table pass through the Buffer to the List-Counter's read port, and accepted track hypotheses are re-issued by the List-Counter for the next pass. (The Buffer is required in order to prevent Holds from propagating back around the loop and stopping the data flow. It is a fast FIFO buffer, capable of performing both a read and a write on each cycle.)

On the second pass, those hypotheses accepted on the first pass are masked with the Y1 hodoscope plane. On this pass, only one of the two Maps is used, having been loaded both with Y1B chamber hits and Y1 hodoscope hits in two pages. The two hit streams are merged using the Switch module (an Arithmetic Operator jumpered to pass data through unmodified, switching back and forth as needed from one input port to the other).

4.3 Transverse Momentum and Mass Calculation

Hypotheses accepted on the second pass are repeated by the second List-Counter for calculation of p_t and mass. On this pass the wire address pairs of good tracks are written into a Block Buffer, to be read out and written to tape along with the rest of the event information; this allows the off-line analysis program to monitor the correct functioning of the processor, and it may also permit the array processor and the analysis program to find tracks more quickly, since the y-view reconstruction need not be repeated.

Actually only the y-component of momentum is computed (since no x information is available to the processor), and we choose to approximate (accurate to 20%)

$$p_t \doteq p_y \ , \qquad m \doteq p_{y1} + p_{y2} \ .$$

At the cost of building a more complex processor we could have included information from the other chamber views and avoided these approximations, but they are adequate for triggering purposes. The set of four Normalizers and two Adders is used once again, this time to compute the quantities θ_0 (production angle in the y-z plane) and 1/p, which, like positions in station 1, are linear combinations of positions at stations 2 and 3.

To compute p_y, these quantities must be divided. It is not worthwhile to design a divider module capable of cycling at 25 ns just for this one use, so we resort instead to subtraction of logarithms. Since these quantities may be either positive or negative, their absolute value must first be taken; this is accomplished using Normalizers, and the signs are recorded in Name bits. The Tables take logarithms to 8-bit accuracy (which is

FERMILAB-CONF-1984-141

adequate for our purpose), which are subtracted using an Arithmetic Operator, and a third Table takes antilog to yield p_y, again with an accuracy of 8 bits.

Since the processor does not use all available chamber information, it does not have the best rejection against accidental tracks, so we must be prepared for events in which the processor finds many tracks, of which only one or two might represent real high-p_t particles. An algorithm for triggering on high-mass events must therefore be able to deal with large numbers of tracks, and to compute the mass of the most massive track pair in each event. To accomplish this we first find the two tracks with the most positive and most negative values of p_y, using the Min/Max modules, then add the absolute values of the two p_y's. This is done separately for dump and target tracks, hence there are four Min/Max modules and two Adders. The calculated p_y and mass values are merged into one data stream using Switches and written into a Block Buffer to be read out with the event, making them available for off-line monitoring.

4.4 Trigger Generation

The ultimate result of the processor's decision is a command issued on the Transport Bus, telling the Block Buffers and other data sources either to read out the event or to skip it and reset the readout system for the next event. This decision is based on the reconstructed p_y and mass values. It is undesirable to reject low-p_y and low-mass events outright, however, since then any bias which the processor might introduce into the data sample can never be corrected. Instead, we prescale events by different factors: events having high p_y or mass are "prescaled" by 1, but only one in sixteen (say) low-p_y and low-mass events are accepted.

This prescaling is performed by the Event Generator Source (EGS), shown at the bottom of Figure 3. The EGS is a Transport Bus module which receives a processor bus input and issues Read and Skip commands on the Transport Bus. It also maintains an event count, which it puts out on the Transport Bus when the event is read out. Its input data stream is a sequence of prescale values terminated by a Complete word. Prescale values may be any power of 2, from 2^0 through 2^{15}. On receipt of Complete, it uses the smallest prescale value of the sequence: if that value is 1, it issues a Read command; if 2, it issues a Read if the low-order bit of the event number is zero; if 4, if the two low-order bits are zero, etc. In this way, fractions ranging from all to 1/32,768 of events can be selected. The prescale values are determined from the calculated p_y and mass values according to the preloading of the two Tables.

We have also implemented a feature allowing the processor to be bypassed; for example, certain classes of study triggers which have already been selected and prescaled by the second-level

trigger logic are accepted regardless of the processor's decision. The leftmost data stream shown in Figure 3 comes from the Trigger Bit Latch system and indicates which second-level triggers fired. If a study trigger fired, the Table sends a prescale factor of 1 to the EGS, forcing the event to be accepted. This also provides a simple way of disabling the processor entirely: the Table can be loaded so that all events are forced through. Since the processor's verdict is still read out, this mode is useful for testing, and we ran the processor in this mode until we established to our satisfaction that our algorithm was working.

5 Current Status

The processor is now debugged and working. At our typical beam intensities the events have many background hits, causing the processor to find many accidental tracks. Some typical events are shown in Figure 4. An average of 10 or 20 tracks are found per event. We are working on improving the accidental-track rejection; options being considered include requiring five out of six chambers per track, using drift time information, and (for muon triggers) using muon proportional tube information.

I wish to thank my collaborators on E605 for their help, and especially Bob Hsiung for his able work in assembling and debugging the processor and its support software.

Postscript (5/14/84): At 10^{10} interactions per pulse and for our typical mix of triggers, the processor reduces the event rate by a factor of 2, with p_y and mass thresholds set at 6 and 8 GeV and prescale factors at 16. The rejection improves to a factor of 5 at 10^9. (These relatively modest factors are due to the already high selectivity of the second-level trigger.) Efforts to reduce the sensitivity of the algorithm to accidentals and improve the rejection are in progress.

REFERENCES:

1. W. Sippach, G. Benenson, B. Knapp, IEEE Trans. Nucl. Sci., NS-27, 578 (1980); H. Cunitz, Y. Hsiung, B. Knapp, W. Sippach, "Data Driven Processing," in Proceedings of the International Conference on Instrumentation for Colliding Beam Physics, SLAC Report-250, June 1982.

2. J.P. Rutherfoord, "On-Line Filtering of High Energy Physics Data With an Array Processor," this conference.

3. J.A. Crittenden, G. Benenson, H. Cunitz, Y.B. Hsiung, D.M. Kaplan, W. Sippach, B. Stern, "A Data Acquisition System for Elementary Particle Physics," to be published in IEEE Trans. Nucl. Sci.

FERMILAB-CONF-1984-141

PROCESSOR BUS FORMAT

24-bit Cable:

Data		Name		Control			
X_{15}	... X_0	N_3	... N_0	C	V	H	B

Control Bits:

V	Valid: here is a data word
H	Hold: couldn't accept that word
C	Complete: end of block
B	Block Reset: abort this block

NOTE: H and B travel against the flow of data, i.e. from receiving module to sending module.

Table 1

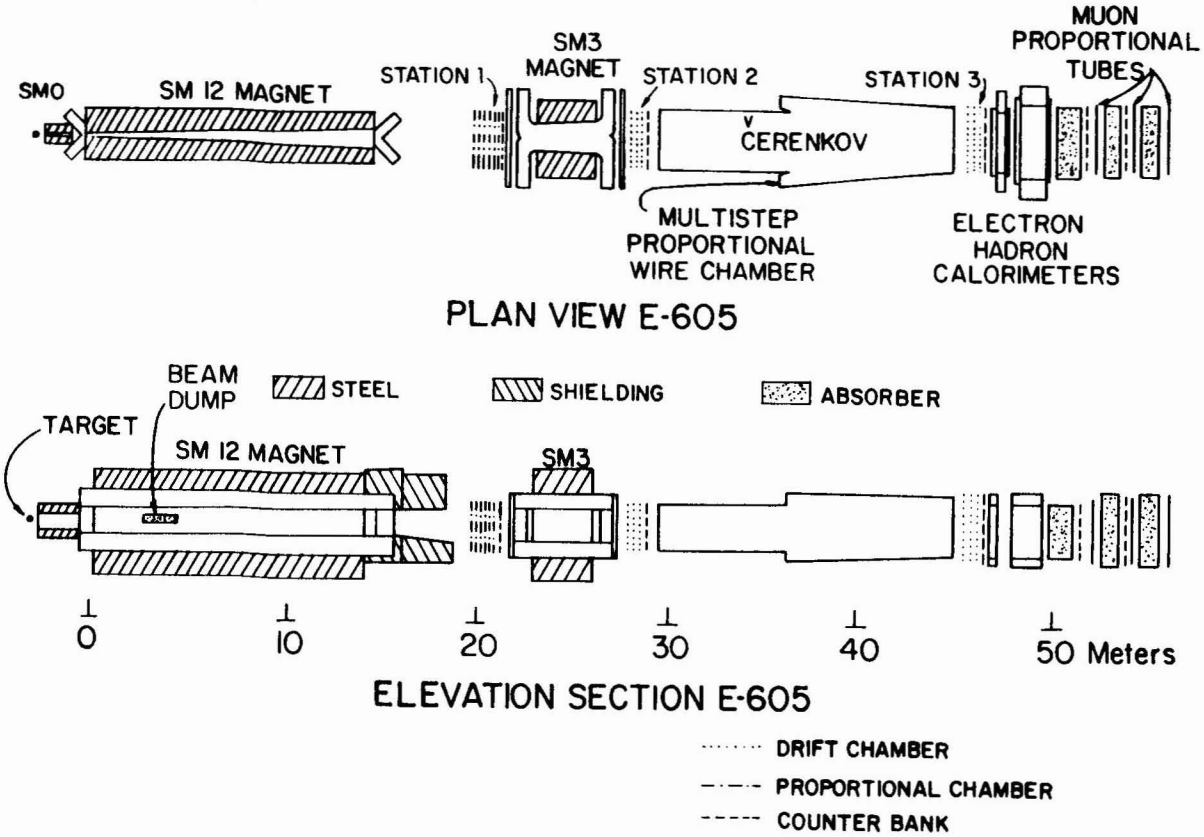
SMO
SM 12 MAGNET
STATION 1
SM3
MAGNET
STATION 2
CERENKOV
STATION 3
MUON
PROPORTIONAL
TUBES
MULTISTEP
PROPORTIONAL
WIRE CHAMBER
ELECTRON
HADRON
CALORIMETERS
PLAN VIEW E-605
BEAM
DUMP
STEEL
SHIELDING
ABSORBER
TARGET
SM 12 MAGNET
SM3
0
10
20
30
40
50 Meters
ELEVATION SECTION E-605
DRIFT CHAMBER
PROPORTIONAL CHAMBER
COUNTER BANK

Figure 1

FERMILAB-CONF-1984-141

Processor Modules

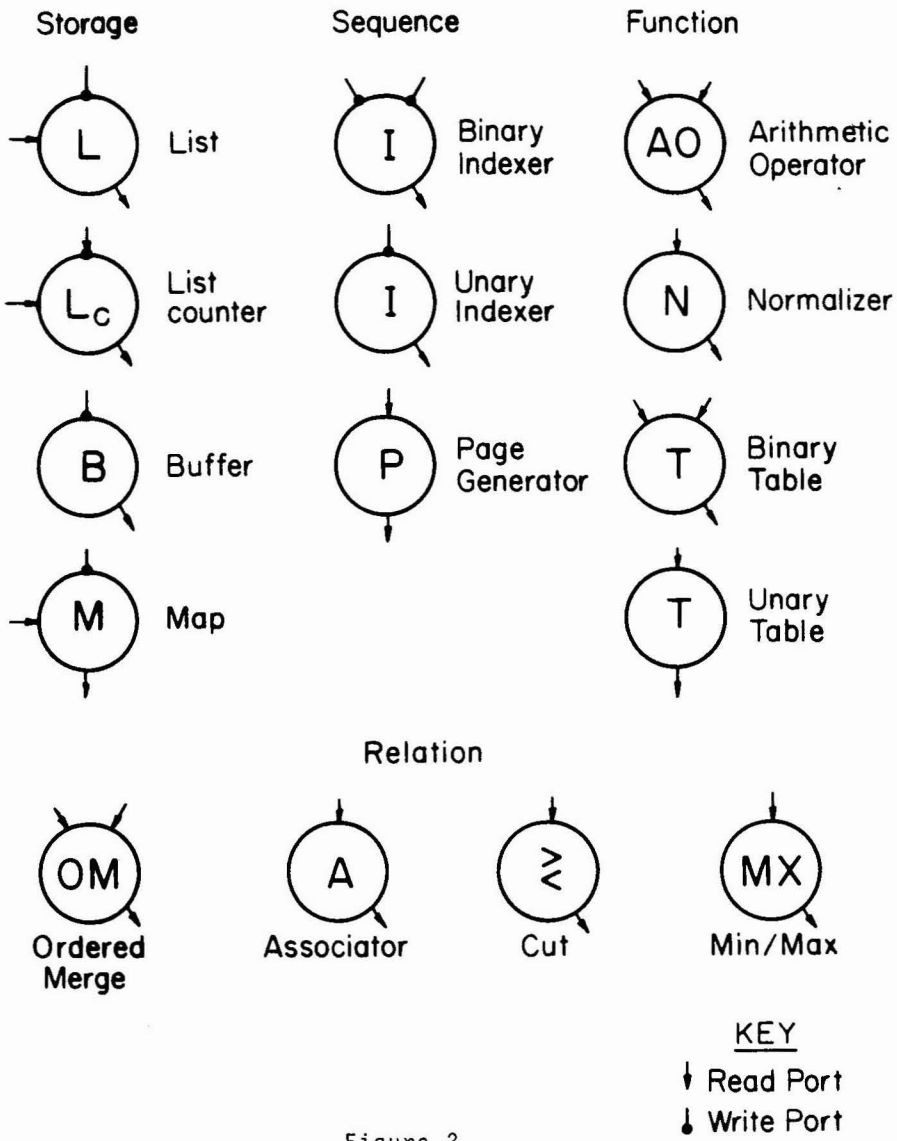

Figure 2

FERMILAB-CONF-1984-141

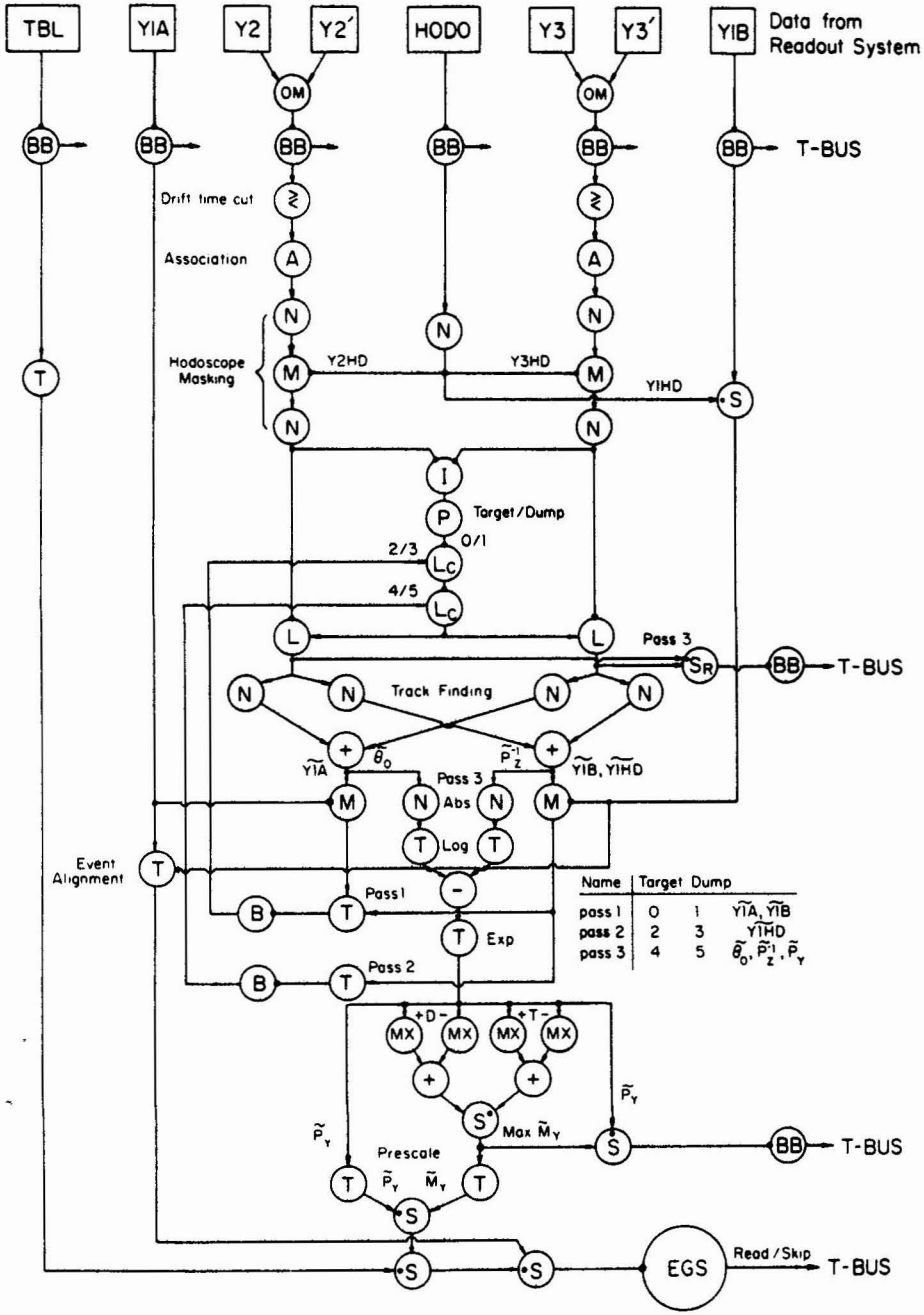

Figure 3: Processor Schematic

FERMILAB-CONF-1984-141

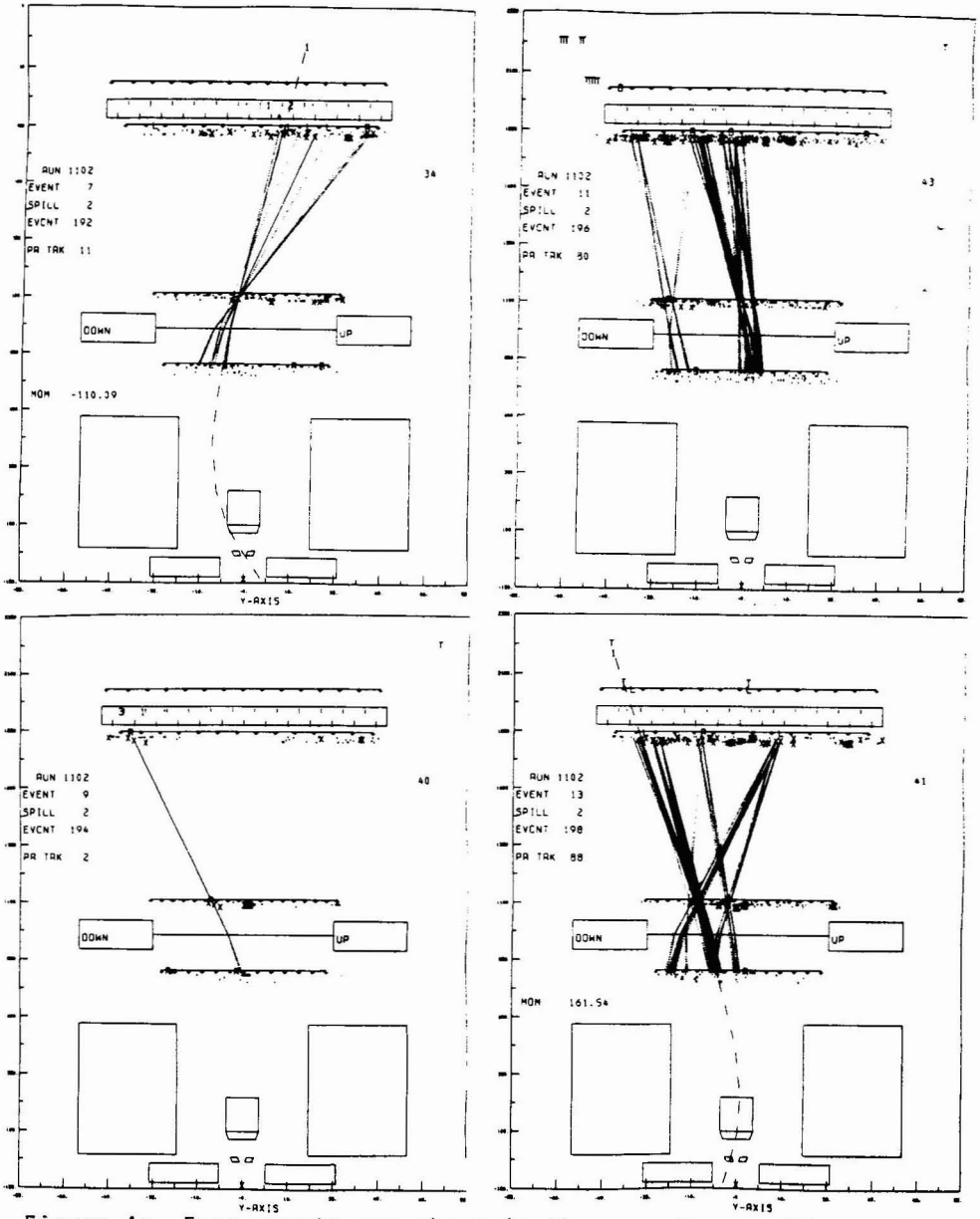

Figure 4: Four events are shown in the y-z plane, with z-axis compressed a factor of 10 relative to y-axis. Tracks found by the off-line analysis are drawn with long dashes, those by the processor with short dashes.